Authentication Schemes for Wireless Sensor Nodes at a Glance

AW1 Report

Tobias Markmann
tobias.markmann@haw-hamburg.de

July 23, 2013

In this work we provide an overview over authentication schemes for nodes of wireless sensor networks. We cover classic authentication concepts, like message authentication codes and public key cryptography, but also visit more recent proposals to authentication problems in distributed networks, i.e. cryptographically generated addresses or identity-based signatures. All of these are discussed with regard to the properties of wireless sensor networks, including energy and power constraints and their open deployment environment. Pairing-based cryptography is shortly discussed, as ways of improving performance of identity-based signatures or allowing new cryptographic concepts.

Contents

1 Introduction

Wireless Sensor Networks (WSNs) have seen a variate of applications in the field of distributed systems. They are used from environmental monitoring to various military applications. WSNs can be characterized as a distributed system of usually low-power computation nodes that collect auxiliary data from their sensors and communicate with their peers in a wireless manner [1, p. 1].

Nodes of a WSN implement three main functionalities: sensing of the environment, aggregation and storage of recorded data and communication between the nodes [2, p. 328]. The communication between the nodes is particular important, because it is the only way for the sensing nodes to move recorded data to a node or machine which will store and analyze it.

The importance of communication between the nodes is making it a critical part of the core functionalities. Allowing secure communication between nodes requires authentication of nodes within the WSN. In addition, the open and energy constrained environment, WSN nodes operate in, set the requirements for the analysis of authentication mechanisms for the nodes.

In most scenarios, data integrity and data origin authentication are the minimum security requirements to prevent modification and insertion of false data into the network, which would otherwise distort the overall results. This can be achieved using Message Authentication Codes (MACs) or cryptographic signatures which are attached to network packets and validated by the receiver. Another approach, using classic Public Key Cryptography (PKC) withPublic Key Infrastructure (PKI), involves a huge key distribution problem on a distributed network of wireless sensor nodes, since every node would need access to the senders' public keys.

In this work we will give a general overview on possible authentication options for the particular constraints and characteristics of WSNs. This includes well established schemes like MACs, classical PKC, i.e. RSA signatures, but also more novel concepts like Cryptographically Generated Address (CGA), Identity-based Signature (IBS) and Attribute-based Signature (ABS). In addition, we do a brief analysis of their viability for use in authenticating nodes in WSNs.

Securing routing protocols, is one possible application of authentication schemes. Due to the limited performance of the sensor nodes, this affords an opportunity for alternative, more energy efficient signature algorithms for use in secure routing protocols for WSNs. This is essential for WSNs, since they are usually battery powered and a high lifetime is required to keep maintenance costs low.

A recent publication showed that IBC is particular suitable for WSNs and compared various IBC signature algorithms for their application in WSNs. However their work was concentrated around pairing-based IBC algorithms [3].

2 Options for Authenticating Wireless Sensor Nodes

There are various ways to authenticate the nodes within a WSN and the messages they send during communication, including communication for routing purposes. This section provides an overview of some ways to implement authentication and puts them in relation to the special requirements of the WSN scenario. The authentication options differ in maintainability, i.e. the work required when new members join the system, efficiency of signing and verification in terms of computing power required, but also in size of the signatures and their attack surface, i.e. gaining information about the all keys of the system when gaining access to a single key.

Since IBSs are a more recent topic in the field of cryptography, compared to PKC, they are described more expansively.

2.1 Message Authentication Codes

MACs provide a way to authenticate messages between a party of communication partners. They enable detection of modification of the message itself, data integrity, but also authentication of data origin, i.e. knowing who send a message. It requires the senders and the receivers to shared a common private secret, the Pre-Shared Key (PSK). Only the parties knowing the PSK can produce valid MACs for messages and are able to verify MACs for messages.

One family of MACs, which have a broad application nowadays, are Hash-based Message Authentication Codes (HMACs), which employ a hashing scheme that can use any cryptographic hash function, i.e. SHA1 or MD5, and turn it into a MAC algorithm [4]. Cipher-based Message Authentication Codes (CMACs) are another option to construct a MAC algorithm. Here an existing symmetric block cipher is used to build a secure MAC algorithm. One example for CMAC is AES-CBC, which uses AES in cipher block chaining mode [5].

The requirement of a PSK between the communication partners has several design consequences for use in protocols and deployment. For one, to identify a single node within a large WSN, you need a unique PSK for each communication subgroup. This results in an unscalable key distribution scenario and each node in the network would still not be uniquely identifiable because of the PSK can be used symmetrically by sender and receiver. Adding a new node to this system would require setting up new shared secrets between the new node and all other members to which communication is intended to, leading to bad maintainability properties of the system.

Another problem, occurring when using a single shared secret for all nodes of a WSN, is the danger of node capture. An attacker gaining access to the shared secret, i.e. by node capture, can send validly signed messages to other nodes of the systems and by that expose the whole network to further attacks.

2.2 Classic Public Key Signatures

PKC is an asymmetric cryptographic concept using different keys for en-/decryption and signing/verification. Some early implementations of this concept are RSA[6], which can used for confidentiality and authentication, and Digital Signature Algorithm (DSA), only for authentication. Each member of the crypto system has it's own private and public key. The private key is used to sign messages and proof the ownership of a certain key. Using the public key, receivers can verify signatures of messages.

To identify nodes in a WSN by their public key, the public key needs to be securely bound to the identity of one particular node and this binding must be known at verification time by the verifying entities. Otherwise they can't know who signed a message. One way to do this, and as it is done in the World Wide Web (WWW), is to use certificates. Certificates basically bind a public key with an identity and are signed by a higher entity, a Certificate Authority (CA), which assures this binding. Using this concept all nodes only have to trust the CA.

There are also PKC schemes, which are based on Elliptic Curve Cryptography (ECC). For the same level of security, ECC-based schemes, like elliptic curve DSA, require smaller public key sizes due to fact that the underlying mathematical problem of DSA, computing discrete logarithms, is much harder on elliptic curves.

Different certificate/key distribution models are imaginable for PKC in WSNs. One way is to distribute all certificates on all nodes. This requires large storing capabilities for the nodes and is hard to maintain on change of membership. Once a node is added to the network, its certificate needs to be distributed to all sensor nodes, so they can identify the new node. Another way of handling the key distribution problem is, sending the certificate, which binds the public key used to create a signature to an identity, along with the message and signature. This certificate, signed by a CA, can then be verified using the static public key of the CA and afterwards, the actual signature can be verified using the public key of the certificate. Since a valid, with respect to the public key in the certificate, signature can only be generated using the secret private key corresponding to the public key, the sender has proven ownership of this private key and is thereby securely identified.

Classic digital signatures, like RSA signatures or DSA, come with high computation requirements and thus are unsuitable for use in the settings of a WSN. However with the advances of ECC, algorithms with the same security level as classic RSA/DSA have been developed, i.e. EC-DSA, which require smaller key sizes and computation time and becoming suitable for some WSN scenarios.

While there are improvements in the area of computation performance, conceptually key distribution is and will remain a problem. Either you pre-distribute and have high maintenance costs, or you send the certificates along with the signatures which adds a noticeable overhead to the communication messages. Communication should be kept short to reduce power usage of the antenna.

Wander *et al.* [7] have shown that, depending on the expected communication scenario, it can be viable to use off-the-shelf PKC with minor adjustments in wireless sensor networks. They suggest reducing X.509 certificates to the bare minimum, the ID, public key and signature, and to use ECC for reduction of key and signature sizes. To establish a shared key between to nodes in a mutually authenticated they use a simplified SSL protocol.

2.3 Cryptographically Generated Addresses

CGAs, as described by Aura [8], provide a way to proof that a public key belongs to a certain communication partner. This is done by having the network address of the communication partner include a hash of the public key. In IPv6 this are the lower 62 bits of the address. CGAs have been primarily designed for authenticating neighbor discovery and router advertisement replies. The public key send along can be proven to belong to the sender by verifying it against the senders address which includes a hash of its public key.

Since CGAs proof ownership of a public key, a CA is not needed. This facilities deployment in distributed and spontaneous settings. However, the CGAs aren't certified themselves and anybody can generate a new valid CGA for a subnet, although resulting in a different address. Having part of the address being occupied for the hash of a nodes public key, limits the free choice of an address [9, p. 83].

CGAs also provide a level or resistance against brute-force attacks, i.e. finding a private/public key pair for which you can generate the same CGA as for an already existing node. This is done using the *sec* parameter, which sets the requirement, that certain amount of bits of the hash of the concatenation of a random value (*modifier*) and the public key are zero.

In the light of the constraints of WSNs, CGAs come with a couple of drawbacks.

An attacker couldn't spoof signatures for other addresses, but using CGAs alone one can't distinguish between an address from a valid member node of a WSN and an attacker since CGAs aren't certified [8, p. 3]. Other signature schemes described in this section, like public key signatures or IBSs, provide certification via either the CA or Trusted Authority (TA). If the feature of certification of membership in a WSN is required, the basic concept of CGAs needs to be extended.

CGAs usually use RSA PKC for signing and verification [8]. RSA computations however are very computation intensive, resulting in quick draining of the battery of wireless sensor nodes. Castelluccia [10] proposed the use of more lightweight signatures schemes within the general concept of CGAs. In particular, Castelluccia [10, p. 232] proposes the use of the MFFS scheme which is higher costs for key generation but signing and verification are more lightweight than RSA. He also evaluated various signature schemes in the context of CGAs for constrained devices, like the nodes within a WSN.

2.4 Identity-based Signatures

IBSs are signatures based on Identity-based Cryptography (IBC), where each party of the system can use any bit string, i.e. an e-mail address or IP-/Ethernet address, as their public key. IBC, first introduced by Shamir in 1985, provides asymmetric cryptography, where an arbitrary string can be used as public key and the corresponding private key is generated by a common trusted entity of the participating entities, usually known as TA [11, p. 47]. The private keys are then securely distributed to each authenticated member of the system. For signature verification only the public parameters of the system, sender's public key, message and signature are needed.

2.4.1 Definition

In any public key scheme each user of the crypto system has two keys, a private key and a public key. The private key, only know to a single user, is used for decryption of messages and signature generation for messages. The public key is used for encryption of plaintext and verification of signatures.

While classic PKI schemes use certificates to bind identities to their public keys, IBC schemes have an implicit binding between the identity and the public key belonging to it.

An ID-based signature scheme is defined as follows:

$$\text{Setup}(1^k) \rightarrow (mpk, msk) \tag{1}$$

$$\text{KeyDer}(msk, id) \rightarrow usk \tag{2}$$

$$\text{Sign}(usk, M) \rightarrow \sigma_M \tag{3}$$

$$\text{Vf}(mpk, id, M, \sigma_M) \rightarrow \{0, 1\} \tag{4}$$

2.4.2 Workflow

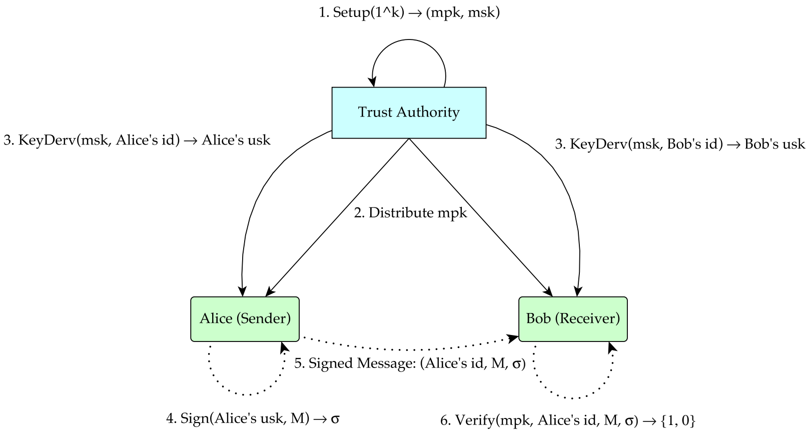
1. Setup(1^k) → (mpk, msk)
Trust Authority
3. KeyDerv(msk, Alice's id) → Alice's usk
3. KeyDerv(msk, Bob's id) → Bob's usk
2. Distribute mpk
Alice (Sender)
Bob (Receiver)
5. Signed Message: (Alice's id, M, σ)
4. Sign(Alice's usk, M) → σ
6. Verify(mpk, Alice's id, M, σ) → {1, 0}

Figure 1: Workflow of a identity-based signature scheme

Setup Figure 1 depicts the process of setting up a distributed system for ID-based signatures. In the beginning the system is initialized with a certain security level with the $Setup$-function by the trust authority (TA). The TA is a central entity controlling the realm for the keys of the system. The *master public key* (mpk) is distributed to all users. The *master secret key* (msk) is kept secret to the TA.

User Key Setup Each user intending to sign messages will need a *user secret key* (usk) which will be bound to the user's identity. The *usk* is generated using the $KeyDer$-function and anyone with access to this key, and and the ability forge her identity, can send validly signed messages as that user.

Signing To sign a message a user (i.e. Alice), uses her *usk* and the message M to calculate σ_M and can then send the ID-message-signature triple to other users.

Verification The receiving party (i.e. Bob) can then verify any ID-M-σ_M triple by passing the *mpk* and the triple as parameters into the Vf-function. Its advantage is that the ID is readily available in most forms of communication anyway.

2.4.3 Characteristics

The authenticity of identities within the system is crucial to the overall security, since the keys (in particular the public keys), are derived from the identity. This reliance on authenticity also enables all parties to verify signatures of any member in the system, without maintaining a dedicated database for keying material for other parties, resulting in a lighter system [12, p. 59f.].

Identity-based crypto systems don't have the key distribution problem, because here the public key needed to verify a signature is derivable from the identity, which in most use cases is readily available to a verifying party, as are all other parameters needed for verification.

Owing to the fact, that the TA generates all private keys in the crypto system, it is able to impersonate all users of the system or in ID-based encryption systems, decrypt all traffic [13].

There are various schemes for realization of IBC, classifiable as either pairing-based or pairing-free. Pairing-based IBC schemes have been researched in-depth during the last 10 years, beginning with the work of Boneh and Franklin, who used pairing-based cryptography to implement an identity-based encryption scheme [14]. Pairing-free IBC schemes haven't seen much attention within the research community compared to paring-based approaches yet. Boneh and Franklin proposed the first pairing-free and space-efficient IBC, which has considerably worse performance [15].

2.4.4 Online/Offline Signature Schemes

One sub area of signature schemes are Online/Offline Signaturess (OOSs), introduced by Even *et al.* in 1990 [16]. Here the signature procedure is split into a computation heavy offline part and a lightweight online part. The offline part is calculated in advance on high performance hardware and the result is then stored on the actual nodes which will be used at runtime. The online function only does little calculations under availability of the actual message to sign.

2.4.5 Discussion

The TA of a ID-based crypto system can be a totally offline entity of the system which is mainly used in the preparation phase of a WSN. The sensor nodes are equipped with the *mpk*, their *ID* and their *usk* in this phase as well which they can store in a secure manner. In many communication scenarios the *ID* is transferred anyway, so the public key doesn't need to be distributed in advance or somehow requested ad-hoc.

Considering that IP addresses or other already existing addresses can be used as public keys, the maintainability is highly increased. This especially suits communication systems, since here the messages/packets already have the sender's address in them. For example, when adding a new node to an existing WSN, you only need to generate a private key for this new node using the common TA. Beyond this nothing is needed and messages send by this new node can be verified by any member node of the WSN.

Typically, classic IBCs systems have a key escrow functionality with the TA generating all private keys. However, a WSN is usually dedicated to a single purpose, deployed by one owner, not sharing the network between the nodes with other vendors. Therefore, the key escrow function of IBC isn't considered to be a problem.

Li *et al.* [3] have shown that signatures based on IBC are practically suitable for WSNs. IBSs based on pairings and ECC have been shown to be especially efficient, so their usage won't drain the batteries of WSN nodes too fast.

2.5 Attributed-based Signatures

ABSs are similar to IBSs, in that the signing party is identified by a set of attributes instead of a single identifying bit string. Respectively to the TA in IBC systems, in ABS schemes there is a attribute-certification authority which issues attribute certificates, similar to the user specific private keys in IBC, to the users [17, p. 60].

An ABS basically shows, that the signing entity possessed some set of attributes, certified by the attribute-certification authority. This way the verifying parties don't know the actual attributes but only the certification of the attributes.

In their paper Li *et al.* propose to add the requirement of non-transferability to ABS-based access control systems, so that attribute proving certificates can't be shared with any other party [17, p. 66]. In the settings of WSNs, where the nodes are usually freely distributed in the environment and might be in a publicly easy accessible area, non-transferability of these certificates is a desirable property since the certificates could be accessed by a malicious party.

3 Pairing-based Cryptography

Cryptography based on bilinear pairings boomed since 2000, when Joux proposed to use bilinear pairings, specifically the Weil-pairing, for a one-round three party key exchange [18]. Pairing-based Cryptography (PBC) not only enabled the implementation of previously suggested ideas for cryptographic schemes, in the case of Identity-based Encryption (IBE) [14], but has also been used to make existing cryptographic schemes more efficient.

The most popular parings over elliptic curves, which have been applied in crypto systems, are the Weil and Tate pairings [19]. Tate parings are more efficiently computable than Weil pairings.

Dutta *et al.* [20] provide a broad overview of cryptographic schemes and protocols which are realised using bilinear parings. This includes various encryption, signature, key agreement and threshold schemes. Short signature schemes, like the BLS signature scheme [21], use pairings to provide small-sized digital signatures compared to classic digital signatures. In the case of BLS, signatures are have the size of DSA signatures for the same level of security. Small signatures are interesting for a WSN scenario, because they help to keep the communication overhead to a minimum.

Szczechowiak *et al.* [22] demonstrate the use of Tate pairings for an identity-based authentication scheme in WSNs. According to them, an identity-based solution addresses the otherwise hard key distribution problem ideally.

4 Conclusions & Outlook

The variety of authentication options for sensor nodes in WSNs mostly differ in maintainability, efficiency and key sizes. While the efficiency, key and signature sizes of PKC-schemes can be improved by use of ECC and of IBS- and ABS-schemes by use of bilinear pairings and elliptic curves, the key usage and distribution patterns can't be as easily changed. CGAs by themselves only protect from spoofing of other nodes and don't provide certified identification. However, this feature could be added with the integration of identity-based concepts.

The key distribution characteristics and its resulting low signature overhead of IBS fit the properties of WSNs very good. Nevertheless, it's important to keep the energy constraints of the wireless sensor nodes in mind when evaluating possible IBS schemes.

Further, it's planned to gain an overview of available ID-based signature algorithms, focusing on pairing-based schemes, followed by an implementation and testing of a suitable scheme for usage in WSN. One possible application of the implementation could be deployed in a system like SAFEST, which aims to improve safety in public places by monitoring for potential hazards [23], to ensure communication authenticity.

References

[1] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks", *Commun. ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004.

[2] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks", *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[3] F. Li, D. Zhong, and T. Takagi, "Practical Identity-Based Signature for Wireless Sensor Networks", *Wireless Communications Letters, IEEE*, vol. 1, no. 6, pp. 637–640, 2012.

[4] M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message Authentication", in *Advances in Cryptology — CRYPTO '96*, ser. Lecture Notes in Computer Science, N. Koblitz, Ed., vol. 1109, Springer Berlin Heidelberg, 1996, pp. 1–15.

[5] J. Song, R. Poovendran, J. Lee, and T. Iwata, "The AES-CMAC Algorithm", IETF, RFC 4493, Jun. 2006.

[6] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems", *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[7] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks", in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, 2005, pp. 324–328.

[8] T. Aura, "Cryptographically Generated Addresses (CGA)", IETF, RFC 3972, Mar. 2005.

[9] J. Arkko, T. Aura, J. Kempf, V.-M. Mäntylä, P. Nikander, and M. Roe, "Securing IPv6 Neighbor and Router Discovery", in *Proceedings of the 1st ACM Workshop on Wireless Security*, ser. WiSE '02, Atlanta, GA, USA: ACM, 2002, pp. 77–86.

[10] C. Castelluccia, "Cryptographically Generated Addresses for Constrained Devices", *Wireless Personal Communications*, vol. 29, no. 3–4, pp. 221–232, 2004.

[11] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes", in *Advances in Cryptology*, ser. Lecture Notes in Computer Science, G. R. Blakley and D. Chaum, Eds., vol. 196, Springer Berlin Heidelberg, 1985, pp. 47–53.

[12] K. Paterson and G. Price, "A comparison between traditional public key infrastructures and identity-based cryptography", *Information Security Technical Report*, vol. 8, no. 3, pp. 57–72, 2003.

[13] K. Hoeper and G. Gong, "Short Paper: Limitations of Key Escrow in Identity-Based Schemes in Ad Hoc Networks", in *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, 2005, pp. 403–405.

[14] D. Boneh and M. Franklin, “Identity-Based Encryption from the Weil Pairing”, in *Advances in Cryptology — CRYPTO 2001*, ser. Lecture Notes in Computer Science, J. Kilian, Ed., vol. 2139, Springer Berlin Heidelberg, 2001, pp. 213–229.

[15] D. Boneh, C. Gentry, and M. Hamburg, “Space-Efficient Identity Based Encryption Without Pairings”, *Foundations of Computer Science, IEEE Annual Symposium on*, pp. 647–657, 2007.

[16] S. Even, O. Goldreich, and S. Micali, “On-Line/Off-Line Digital Signatures”, in *Advances in Cryptology – CRYPTO' 89 Proceedings*, ser. Lecture Notes in Computer Science, G. Brassard, Ed., vol. 435, Springer New York, 1990, pp. 263–275.

[17] J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren, “Attribute-based Signature and its Applications”, in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '10, Beijing, China: ACM, 2010, pp. 60–69.

[18] A. Joux, “A One Round Protocol for Tripartite Diffie-Hellman”, in *Algorithmic Number Theory*, ser. Lecture Notes in Computer Science, W. Bosma, Ed., vol. 1838, Springer Berlin Heidelberg, 2000, pp. 385–393.

[19] ——, “The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems”, in *Algorithmic Number Theory*, ser. Lecture Notes in Computer Science, C. Fieker and D. Kohel, Eds., vol. 2369, Springer Berlin Heidelberg, 2002, pp. 20–32.

[20] R. Dutta, R. Barua, and P. Sarkar, *Pairing-Based Cryptographic Protocols : A Survey*, Cryptology ePrint Archive, Report 2004/064, `http://eprint.iacr.org/`, 2004.

[21] D. Boneh, B. Lynn, and H. Shacham, “Short Signatures from the Weil Pairing”, *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[22] P. Szczechowiak, A. Kargl, M. Scott, and M. Collier, “On the Application of Pairing Based Cryptography to Wireless Sensor Networks”, in *Proceedings of the Second ACM Conference on Wireless Network Security*, ser. WiSec '09, Zurich, Switzerland: ACM, 2009, pp. 1–12.

[23] E. Baccelli, L. Gerhold, C. Guettier, J. Schiller, T. C. Schmidt, G. Sella, U. Meissen, A. Voisard, M. Wählisch, and G. Wittenburg, “SAFEST: A Framework for Early Security Triggers in Public Spaces”, in *Proc. of WISG 2012 – Workshop Interdisciplinaire sur la Securite Globale*, Troyes, France, Jan. 2012.