2015 Euromicro Conference on Digital System Design

Towards the use of Pairing-Based Cryptography for Resource-Constrained Home Area Networks

Rune Hylsberg Jacobsen and Søren Aagaard Mikkelsen
Department of Engineering, Aarhus University, Denmark
Email: {rhj,smik}@eng.au.dk

Niels Holm Rasmussen
Tieto DK A/S, Aarhus, Denmark
Email: niels.rasmussen@tieto.com

***Abstract*—In the prevailing smart grid, the Home Area Network (HAN) will become a critical infrastructure component at the consumer premises. The HAN provides the electricity infrastructure with a bi-directional communication infrastructure that allows monitoring and control of electrical appliances. HANs are typically equipped with wireless sensors and actuators, built from resource-constrained hardware devices, that communicate by using open standard protocols. This raises concerns on the security of these networked systems. Because of this, securing a HAN to a proper degree becomes an increasingly important task. In this paper, a security model, where an adversary may exploit the system both during HAN setup as well as during operations of the network, is considered. We propose a scheme for secure bootstrapping of wireless HAN devices based on Identity-Based Cryptography (IBC). The scheme minimizes the number of exchanged messages needed to establish a session key between HAN devices. The feasibility of the approach is demonstrated from a series of prototype experiments.**

***Index Terms*—home area network; constrained devices; security; network bootstrap; pairing-based cryptography; identity-based cryptography.**

I. Introduction

The use of gateway-connected sensors and actuators deployed for smart grid operation as part of a HAN are often based on low-power wireless radio platforms with limited processing capabilities, memory, and power available forming Wireless Personal Area Networks (WPANs) [1]. The hardware platforms utilized for these HAN devices are typically based on 8-bit or 16-bit microcontrollers optimized for low-energy operation by supporting sleep modes of the microcontrollers and thereby allowing devices to run duty-cycled with a ratio between awake state and sleep state of about 1/1000 or less. For instance, microcontrollers like the MSP430 and the ATMega128 are very popular choices for HAN devices since these have good power saving features and come at a reasonable cost.

When HAN devices participate in a Home Energy Management System (HEMS) for a smart grid operation they provide the bi-directional communication channel needed to monitor and control appliances in the household. These devices need to be authenticated so that the electricity provider company is guaranteed the validity of information from the devices.

Traditional authentication methods use a Trusted Authority (TA) or a Certificate Authority (CA). However, the traditional methods are not suited for device authentication in a HAN environment due to its resource-constraints. The method of using certificates and trusted third parties is too reliant on connectivity, processing power, and bandwidth to uphold a Public Key Infrastructure (PKI) [2]. In addition, it becomes a cumbersome task to keep every device updated with recent certificates for large HANs.

The traditional approach of key distribution in WPANs relies on predefined keys established at time of device provisioning. In contrast, the HAN scenario presented here enables the devices and the HAN gateway to authenticate each other by using data provided by the end-user. This follows the *transitive trust principle* [3]. A secure boot is a mechanism to apply the transitive trust principle. The gateway will trust the device, because the user has provided a functional trust to the device. A particular challenge arises during the early stages of the HAN device life cycle when the devices transit from initial field deployment to the start of the security bootstrapping [4]. Security bootstrapping includes the authentication of devices to establish trust relationships in the HAN, as well as to transfer security parameters and keying materials. Once secure and authentic communication channels are established, the bootstrapping of all other information can be carried out as ordinary secured communications. At first, an initialization key must be established. This can be accomplished by using predefined keys uploaded to the devices at manufacturing time or by allowing the user to enter a key at installation time. Both methods pose a breach of security of the system. As an example, when distributing the master key in a ZigBee network, ZigBee devices are allowed unencrypted over-the-air transport [5]. The risk involved is explicitly addressed in the ZigBee specification [6]: "If the applications can tolerate a moment of vulnerability the master key can be sent via an in-band unsecured key transport." This operation results in a vulnerable joining phase and leads to a network that is easy to compromise. These weaknesses have been exposed by the Killerbee software framework and tool set for exploring and exploiting the security of ZigBee and IEEE 802.15.4 networks [7].

This paper presents and evaluates a method for authentication of resource-constrained devices in a HAN acting as communication infrastructure in the customer premises of a smart grid. It applies a traditional PKI in which the HAN gateway provides a public key to end-devices. For device-to-device authentication, the proposed method uses Elliptic Curve Cryptography (ECC) to derive a symmetric key from devices'

978-1-4673-8035-5/15 $31.00 © 2015 IEEE
DOI 10.1109/DSD.2015.73

Identifications (IDs) based on Pairing-Based Cryptography (PBC) techniques [8]–[10]. For this authentication step, only the exchange of device IDs is needed since devices are able to generate a shared key using their own private key and the ID of another device. The technique is known as Identity-Based Cryptography (IBC) and it requires minimal user involvement [11]. Trust to a new device is transferred from the user to the device managing the network i.e., the HAN gateway. A set of prototypes are developed to evaluate the feasibilty of the proposed authentication method with respect to processing time, the power consumption, and the added payloads on the limited and costly bandwidth.

The remainder of the paper is organized as follows: Section II describes related work. Section III introduces our security model. Section IV presents the basics of PBC. In Section V, we propose a secure and efficient HAN bootstrapping mechanism. This is followed by a description of our prototype implementations in Section VI. The evaluation of the mechanism is found in Section VII followed by a discussion in Section VIII. Finally, we conclude the paper in Section IX.

II. Related Work

Until recently it was a common belief that resource-constrained devices would not take part in Public Key Cryptography (PKC) due to the need for large key sizes [12], [13]. The problem of security, message integrity and device authentication, can in most cases be solved by using a strong encryption scheme. Current recommendation by the National Institute of Standards (NIST) states that at least a 128-bit key strength is needed for most of today's applications. Alternatively, keys with low bit-strength may be used with a frequent replacement to hinder the communication from being compromised. However, to achieve a comparable level of security with PKC much longer keys are needed. For example the security available with a 1024-bit key using asymmetric Rivest-Shamir-Adleman (RSA) cryptography is considered approximately equal in security to an 80-bit key in a symmetric algorithm.

Recent developments within ECC have opened up for the possibility of implementing a PKC for resource-constrained devices. In contrast to RSA, that utilizes the hardness of the integer factorization problem, ECC benefits from the discrete logarithm problem for providing hardness in its security operations. The technique is based on the algebraic structure of elliptic curves over finite fields which was introduced in 1986 [14]. The two methods differs in the way a key is derived and in the size of keys that are needed to achieve the same levels of bit security [15]. This results in a much faster encryption and decryption speed in contrast to the large keys used in RSA, making ECC-derived public keys interesting for constrained devices [16].

A number of available ECC implementation are suitable for resource-constrained devices. Three such implementations are EccM 2.0 [2], TinyECC [17] and TinyPBC [9]. When implemented on a resource-constrained device, ECC needs optimization. However, it still ends up with execution times on the order of seconds with reasonable key sizes [12]. This is considered too high to be applied as the basis for key management during the steady operation of a HAN. Nevertheless, ECC can be used for one-time authentication of a new device without a key management protocol since a booting time interval of few seconds can be tolerated. TinyPBC is an IBC implementation providing the basis for PBC [9]. This means that public keys are not exchanged between devices, only the exchange of device IDs is needed. The software can derive a 160-bit symmetric key between two parties without exchanging other information than device IDs, provided a trusted *key generator* has distributed private keys beforehand. Instead of having a network-shared public key, TinyPBC requires the distribution of private keys derived from a *master secret* (a system-wide secret key). TinyPBC runs on the TinyOS operating system suitable for resource-constrained devices [18]. EccM and TinyECC have potential as methods for providing a PKI in HAN environment, as they are independent of a shared linchpin secret, such as the master secret in TinyPBC. However, they both need some rewriting to be compiled in TinyOS 2.x and there is furthermore potential for optimizing the ECC operations. TinyPBC has recently undergone an optimization process reducing the execution time from 5.5 seconds [19] to 1.9 seconds [9]. EccM and TinyECC are some years older than TinyPBC; this being the reason for the earlier TinyOS 1.x implementation.

Nicanfar *et al.* [20] have proposed an authentication and key management mechanisms for HANs. The authors provide the device and their trust agent with unique symmetric information used for authentication, and asymmetric keys generated by an IBC scheme. The key generation scheme of their system utilizes an ID-based cryptography architecture [21]. This allows devices to calculate the public key of the HAN from knowing the HAN ID. Subsequently, it encrypts messages destinned for the HAN's trust agent instead of requesting the public key, hereby saving on communication overhead. By encrypting the senders' device ID and a unique session key and broadcasting it, the HAN trust agent can decrypt the message and reply using the session key for encryption. The approach by Nicanfar *et al.* [20] requires the configuring of both the end-device and the trust agent with information during installation time. For most consumer applications, this is not practical because the broadcasted messages must reach the trust agent requiring all devices to be within one hop of the trust agent or alternatively rely on all devices to uncritically relay messages and thereby opening up for denial-of-services attacks.

III. Security Model

In this paper, an architecture and security model similar to the model provided by Hjort and Torbensen [22] is considered. Fig. 1 shows the high-level structure of the system. The HAN is assumed to be able to provide security only on subnetwork level and it must implement a TA function. This means that messages coming from the HAN devices must be decrypted in the subnetwork adapter before the HAN gateway can forward the message over a Tier 2 network to an IP network

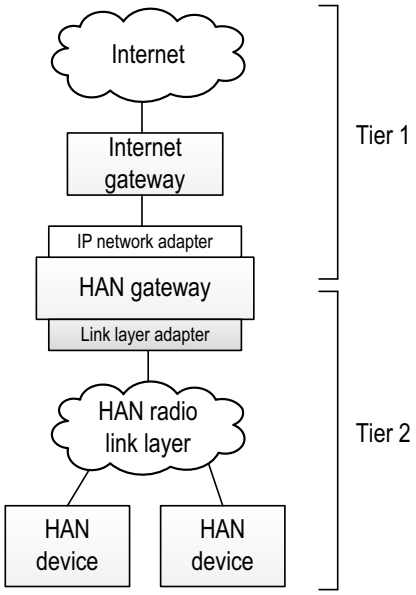

Fig. 1. Structural model of the HAN system.

(Tier 1) e.g., over a wired or wireless Local Area Network (LAN) technology implemented in the home such as e.g., Ethernet or WiFi. The Tier 1 network is assumed to provide its own security system such as IPSec or Transport Layer Security (TLS). Since the HAN gateway is more powerful that other HAN devices, it can implement more strict security requirements. Furthermore, most HAN technologies are based on the concept of a master device or a coordinator node [6]. End-devices only need to trust the master device in order to be reachable from Tier 1, and the HAN gateway essentially acts as a Tier 1 controller device in the subnetwork since it relays messages transparently.

The gateway is assumed to implement an end-device Access Control List (ACL) for the HAN. It is asumed that the user, i.e., the HAN owner, that controls the members of the ACL. However, there is the danger of the user being interested in tampering with data reported to the electricity provider company. However, the protection against tampering attacks is beyond the scope of this paper.

Due to the nature of HAN devices, some may need to be put in places with low physical security such as embedded in outdoor installed smart meters where they can be accessed by unauthorized persons. Therefore, HAN devices should not have permission to reconfigure other end-devices in the network. Only controllers and HAN gateways on Tier 1 should be authorized to send reconfigurations to devices. This ensures that even if an end-device is compromised, it is only permitted to make its functionality available to the particular controller that it may may choose to utilize, and it is therefore fully segregated from the IP network.

The security model assumes that an adversary is present during the complete life-cycle, i.e., from device provisoning to decommissioning. The adversary could for instance be residing in a neighbor apartment of a multi-tenant building. From this location, the adversary is able to eavesdrop messages in the HAN both during bootstrapping and during operations. Furthermore, in a replay attack the adversary can record the flow of messages and reuse these messages in order to successfully get authenticated. Hence, the system must provide protection against replay attacks.

IV. Pairing-Based Cryptography Background

The concept of generating keys based on device IDs is known as IBC [11]. An identity-based encryption scheme enables the deployment of a PKC system without the prior setup of a PKI. The implementation of IBC provides the capability of calculating public keys for devices by using a *master secret* and the unique identities of other devices. Instead of using certificates, IBC provides authenticity of the key from a mathematical technique known as bilinear pairing.

The major pairing-based construct in PBC is the bilinear map formed by the Cartesian product between two groups. If a pairing of a group G_1 is done with the same group it is known as a self-bilinear map. It can be expressed as:

$$e : G_1 \times G_1 \rightarrow G_2 \tag{1}$$

where e is the mapping function which is assumed to exist. G_1 is a cyclic (abelian) group of prime order q written using additive notation and G_2 is a cyclic group written using multiplicative notation. Fortunately, it is possible to find G_1 and G_2 where these properties hold. In particular, the Weil and Tate pairings prove the existence of such constructions [2], [8].

Consider a HAN with N communicating devices with unique identities ID_x where $x \in \{1, 2, \ldots N\}$. These device identities could for example be the link-layer addresses, globally unique IP addresses, or unique domain names. Let the devices construct a cryptographically hashed representation of their identity:

$$P_x = H(\mathrm{ID}_x)\,, \tag{2}$$

where $H : \{0,1\}^* \rightarrow G_1$ is a hashing and mapping function and P_x is a generator of the group G_1. Each device x has an associated private key, denoted S_x, that is derived from a master secret s. The master secret belongs to the set of integers modulo q ($s \in \mathbb{Z}_q^*$), and it is only known by the TA in the HAN. The TA computes a private key S_x of a device x from its hashed identity P_x by:

$$S_x = sP_x\,, \qquad \text{where } S_x, P_x \in G_1 \ . \tag{3}$$

The private key is distributed to the device x that it is generated for. Due to the hardness of the discrete logarithm problem, it is infeasible to calculate the private key from knowledge of the device ID as long as the master secret is kept secret. The private key S_x is used for encryption/decryption and it is not shared with any other devices in the HAN. However, the TA has knowledge of all private keys in the HAN and is therefore able to decrypt any communication between HAN devices.

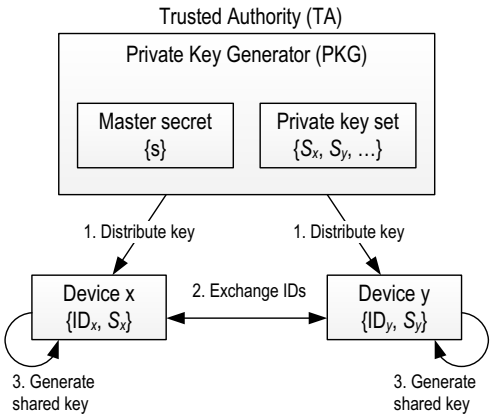

Fig. 2. Identity-based key distribution mechanism.

Pairings in ECC are functions which map a pair of elliptic curve points to an element of the group that arises from the bilinear mapping. In the Weil and Tate pairing, e is a bilinear and non-degenerate pairing of points on an elliptic curve $E(\mathbb{F}_q)$ over the finite field $\mathbb{F}_q$ of q elements [8], [23]. The shared key between two HAN devices results from this mapping by providing the elliptic curve algorithm with a pair of points on the elliptic curve. This is a result of the bilinearity property of the mapping function. For two devices, $x \neq y$, the following must hold for any element $a \in \mathbb{Z}_q^*$:

$$\begin{aligned} K_{xy} &\equiv e(S_x, P_y) = e(aP_x, P_y) = e(P_x, P_y)^a \\ &= e(P_x, aP_y) = e(P_x, S_y) \end{aligned} \quad (4)$$

with x, y being device identifiers. Eq. (4) states that the private key of x combined with the hashed identity of y yield the same point on the elliptic curve as the private key of y combined with the hashed identity of x. Hence, this results in a shared secret key K_{xy} between device x and y.

To summarize, Fig. 2 shows the concept of identity-based key exchange for two devices x and y. First, the TA to generate the private key set in the Private Key Generator (PKG) function and distribute these keys to end-devices (step 1). Second, end-devices exchange IDs with devices within radio range (step 2) by using a neighbor discovery mechanism. Finally, end-devices calculate their shared secret keys K_{xy} for communication with neighbor devices (step 3).

To setup authentication between the HAN gateway and the end-devices, the TA creates a public key as follows: It picks a random generator $P_r \in G_1$ and publishes $K_{TA} = (P_r, sP_r)$, where s is the secret known from Eq. (3). End-devices can use this key for encrypting messages sent to the gateway.

V. Design of HAN Bootstrapping Protocol

Several logical steps are required by HAN devices to follow before these can enter the security bootstraping. First, devices need to hardware boot and associate with a WPAN. This includes establishing networking parameters such as an IP address. Second, routing information needs to be established to define and maintain paths from individual devices to the rest of the network. Third, Security Associations (SAs) should be established with the relevant entities of the network. Hereafter any applications, that should run in the HAN, need bootstrapping including registration and activation of services and the establishing of SAs on the application level.

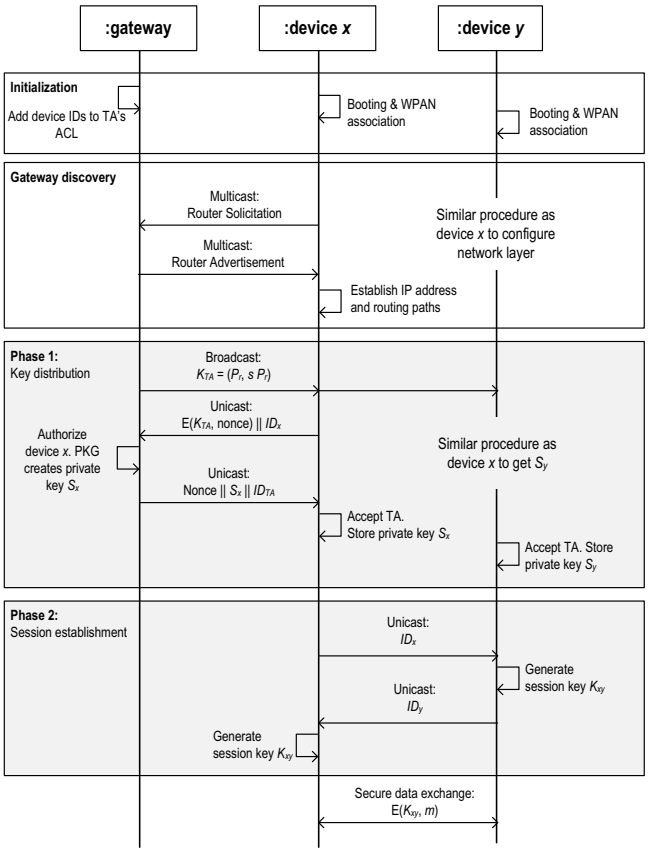

Fig. 3. Secure bootstrapping protocol for the HAN. Protocol messages in white and grey rectangles are accountable for normal and secure booting, respectively.

Fig. 3 shows the proposed authentication method for the HAN bootstrapping with a gateway acting as TA and two end-devices x and y. The authentication method can easily be extended to more devices. It is assumed that there is only one gateway in the HAN. Moreover, it is assumed that the user has provided the needed information to construct the ACL in the gateway to be used by the TA. For simplicity, the establishment of basic SAs is not shown in the figure.

The proposed authentication method essentially divides into two phases. Phase 1 concerns initial key distribution and the TA's authorization of end-devices. In phase 2, end-devices establish secure sessions with neighboring devices.

A. Key Distribution in the HAN

In the first part of the secure bootstrapping, end-devices in the radio range of the gateway are authenticated by the TA and private keys are subsequently delivered to the HAN devices (Fig. 3, phase 1). The authentication process uses asymmetric encryption and requires that the user has provided

trust to the gateway by entering IDs of trusted end-devices to the TA and thereby configuring the ACL. As an end-device boots it starts to look for a valid gateway by using a neighbor discovery mechanism such as IPv6 Neighbor Discover Protocol (NDP) [24]. A device uses the Router Solicitaiton (RS) message of NDP to solicit information for IP address configuration and routing path information, which is sent from the gateway to the device by using the Router Advertisement (RA) message of NDP. The reciept of a RS message is assumed also to trigger a multicast of the public key K_{TA} of the HAN gateway to all configured devices, i.e., by using the All-nodes multicast address. Alternatively, the distribution of the public key can be send as periodical multicast or broadcast messages.

When the gateway receives a message with a device ID, it checks if it holds authentication information related to this device. Besides the device ID, this message consists of a *nonce* and possible other session data. The nonce is encrypted by the public key of the gateway and hence provides a mean for devices to authenticating the gateway, i.e., to validate the authenticity TA. The nonce is furthermore intended to protect against replay attacks. The TA subsequently runs the PKG function to derive the private keys of end-devices. The PKG must verify the identity of devices before delivering the private keys.

B. Session Establishment Between HAN Devices

Phase 2 of the secure bootstrapping concerns the establishment of secure communication between end-devices. It relies on trusted information provided by the TA. Trust is delivered by the private keys derived from the device IDs and delivered by the TA. HAN devices may use the NDP to discover neighboring devices [24]. Subsequently, device IDs can be exchanged between neighbors by using link layer communication. Hereafter, the devices calculate the shared session keys K_{xy} for mutually paired sessions in the HAN. HAN devices are now fully commissioned and ready to operate securely.

VI. Prototype Implementations

To validate the feasibility of using ECC in resource-constrained HAN devices, a series of prototypes were made. Initially, simulations are performed to estimate the number of cycles necessary for booting. Subsequently, a prototype implementation for device bootstrapping with TinyPBC is made. The objective of this prototype is to determine the timing and the processing power required for the proposed security operations. In addition, a study on the energy usages is provided. Finally, work towards a HEMS prototype for a real field deployment is presented.

A. Simulations with Avrora

As a microprocessor cycle metric exists for the symmetric key generation of the bootstrapping protocol on a MICA2 platform [9], the amount of cycles needed to implement the same function here is simulated using Avrora [25]. Avrora is an open-source cycle-accurate simulator for embedded implementations using the MICA2 or MICAz platform; the predessor of the IRIS platform.

B. Bootstrapping with TinyPBC

The software implementations are made in TinyOS 2.1.1 on an IRIS platform from Memsic (www.memsic.com). The IRIS platform consists mainly of an 8-bit ATmega1281 microcontroller running at 7.37 MHz with an AT86RF230 radio transciever. The transceiver supports the IEEE 802.15.4 low-power radio standard [26]. No hardware acceleration for encryption is supported by the platform. The entire platform is powered by two AA batteries. The prototype implementation builds on a variation of IBC called TinyPBC [9]. The software derives a 160-bit key based on ECC without exchanging other information than device IDs. TinyPBC is available online: sites.google.com/site/tinypbc/. The TinyPBC software is based on Relic cryptographic toolkit which is also available: code.google.com/p/relic-toolkit/.

In the prototype implementation, the TinyPBC software is encapsulated into two main functions. The first function sets up the device's ID, initializes the Relic toolkit with function call "*core_init*"and derives the private key from the master secret and the device ID. The second function "*agreeKey*" calculates the shared key between two devices by using the private keys of devices, its own ID and the receiver device's ID. The calculations are executed in the "*cp_sokaka_key*" function, using ECC calculations supported by the Relic toolkit.

Some adaptations must be done for the TinyPBC code to be used by the prototype application. Before the key is useful for encryption purposes, it must be truncated or hashed to a desired bit length [9]. The key will be usable for symmetric encryption. Measuring of software execution time values is executed by toggling an output pin on the IRIS platform [27] at each end of the operation that is being measured, and the time period in between the two toggles are recorded with a digital oscilloscope (Agilent DSO6014A). The time for executing the "*cp_sokaka_key*" function on the IRIS platform is measured with an oscilloscope probing an output pin, connected to a LED on the IRIS platform. The time between the two voltage changes is measured and the execution time is derived from this value. This value does include a small, systematic error as the execution of the LED toggle adds a few processor cycles. Measurements are done using a 10 Ω shunt resistor. An example of a recorded trace is shown in Fig. 4.

To evaluate the performance of the implementation, a set of relevant measurements for this scenario is defined. These measurements include the added overhead for the initialization of the Relic tool chain needed to execute the ECC operations; the time it will take for a gateway to generate a private key; and the time it will take for a device to generate a shared key.

C. HAN Prototype

In the European FP7 research project SmartHG we use the hardware platform for the HAN installation shown in Fig. 5. The HAN gateway consists of a Raspberry Pi module equipped

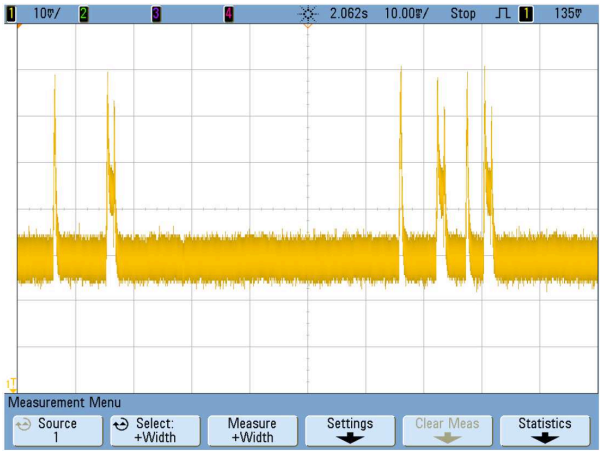

Fig. 4. Example of a trace of the voltage of the IRIS mote. The mote is set to transmit 32 bytes of payload continously.

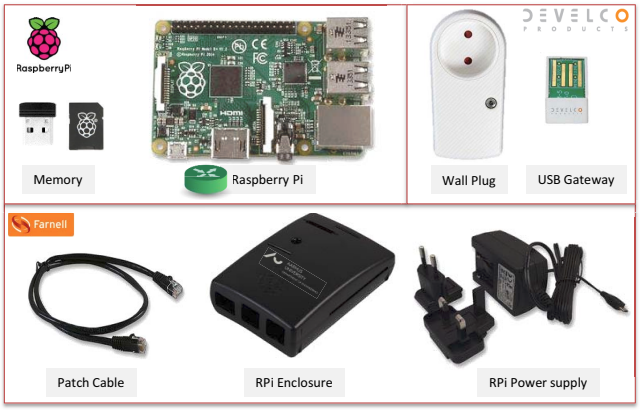

Fig. 5. Basic HAN hardware installation kit.

with a ZigBee bridge. One or more ZigBee smart plugs are part of the HAN which can be further expanded with passive infrared sensors, and temperature sensors, etc. connected over ZigBee. The smart plug acts as a submeter and a actuator switch relay which allows the HAN to support smart grid services. The HAN gateway connects to an IP network either using WiFi or by using Ethernet. This prototype is currently being extended with security functions and is planned for deployment in a smart grid pilot. The gateway run the Linux operating system.

VII. Evaluation

Most evaluations are done with the IRIS platform [27]. The IRIS platfor is the successor of the MICA2 or MICAz platforms are also partlyaddressed.

A. Evaluation of Bootstrapping

Table I shows measured and calculated values for the IRIS platform. The generation of a shared key adds no additional messages. The updating of a private key adds twice the overhead of updating an AES 128-bit key, private key in

TABLE I
Measured and calculated values for voltage, current and power.

Operating mode	U_{meas} [mV]	I_{data} [mA]	P_{data} [mW]
IRIS, sleeping	0.5 ± 0.19	0.054	0.160
IRIS, full operation	89.0 ± 1.42	8.897	26.592
Radio on, μC sleeping	175.2 ± 0.87	17.524	52.377
Radio transmitting (1 mW)	175.8 ± 1.25	17.581	52.547

Values are given +/- one std. deviation. U is the voltage, I is the current and P is the power.

TinyPBC is 272-bit key size [9], and AES with 128-bit key size.

The proposed bootstrapping protocol ends up in a total of two private key and two symmetric key function executions. According to the TinyPBC developers, the time for executing the most time consuming operation, the symmetric key generation, is 140 ms [9] on a Imote2 platform with a PXA27x processor running at 13 MHz. This means that the overhead will be around 300 ms for a gateway with the same hardware specification as the Imote2.

The device receives the authentication message from the gateway. It will need to generate a symmetric key to decrypt the message containing the new private key, read out the new key, and generate a session key for later communication before being fully commissioned. The processing overhead of the device ends up in a total of 4240 ms added to the device boot time, Relic initialization and time used for device discovery. Furthermore, time must also be added for receiving the 272 bit size new private key. However, this is considered to be an acceptable one-time delay.

TinyPBC should be able to achieve the execution of the key pairing within 1.90 seconds on a MICA2 with an ATMega128L microcontroller [9], which is similar to the IRIS with the ATMega1281V. Despite this, the measured execution time on the IRIS is 2.12 seconds and the resulting cycles from a simulation in Avrora on a MICA2 is 15.6×10^6 cycles. This is about 11% higher than the result of 14×10^6 cycles, presented by the developers. The 11% difference is the same in the timing measurements.

The energy consumed in a device to complete joining and authentication protocols and receive its new private key before it enters its operational phase, is derived by calculating the energy cost of the individual operational cost ΔE_{op}. The values are calculated by multiplying the operations time, Δt_{op}, with the power cost of doing operations on the microcontroller, ΔP_{op}:

$$\Delta E_{op} = \Delta t_{op} \cdot \Delta P_{op} \tag{5}$$

Table II shows the energy cost at various steps in the commissioning phase. Overall, the measurements show that the energy consumption is approximately 48% higher that the consumption calculated using values from the datasheet of the IRIS mote [27].

TABLE II
ENERGY COST CALCULATIONS AND MEASUREMENTS DIVIDED BY THE DIFFERENT MODES OF OPERATIONS.

Operation	Theoretic cost [mJ]	Measured cost [mJ]
Relic initialization	4.716	6.967
Shared key generation cost	38.159	56.376
Private key generation cost	4.698	6.941
Device setup (4240 ms)	76.318	112.752
Distribution of 128-bit private key	0.360	0.259

VIII. DISCUSSION

TinyPBC was chosen in the implementation of the second prototype. The key generated from TinyPBC is a 160 bit key derived by ECC. Traditionally, an ECC key of this length, will have a bit security of 80 bit. This rating is for asymmetric encryption and how the key will be rated, if it is truncated to 128 bit or hashed to 256 bit and used in AES is for further study. The AES encryption should still rate at its key length, but the key is derived from an algorithm with a lower rating.

A question, that rose during the analysis of device authentication methods, concerns the energy cost for encrypting the data in a network. To answer this question, there are some parameters to consider. What kind of encryption and what kind of data are being encrypted? The cost of encrypting the data, with the consumption scenario, is a loss in system lifetime of 0.005%-0.010%, depending on theoretic or measured power consumption data, the theoretic consumption being the highest loss. Looking at this low power overhead added and the typical threats where several of them can be combated by encryption, it is clear that encryption is a necessity in a HAN, as sensitive data is transmitted. The bit security of 128 bits may seem a bit high today. However, it only follows the current recommendation from NIST [28] saying 128-bit keys should keep data safe to around year 2030, whereas 80-bit security was only estimated as save to the year 2011. Finally, the question on revocation of trust from devices leaving the HAN needs attention. This is, however, beyond the scope of this paper.

Key distribution is one of the most basic problems in cryptography. Frequently refreshed keys are needed for encryption algorithms and Message Authentication Codes (MACs) to provide confidentiality and integrity security services. Two parties, x and y, who want to establish a shared key K_{xy} may not be able to afford to engage in a Diffie-Hellman (DH) protocol because the processing of long cryptographic keys requires high processing power and hence it is a too strenuous activity for resource-constrained devices.

The symmetric key derived from the exchange is not stored forever. Depending on its strength and usage, it is given a certain lifetime. The key can for example be classified as a session key, where it is destroyed at the end of the communication session involving the two parties or be given an even longer lifetime of up to several years if regular and extensive communication occurs. It is possible to revocate trust of a single device and achieve a full key management scheme wWhen a HAN trust domain [22].

The NDP [24] and its optimization for IPv6 over Low-Power Wireless Personal Area Networkss (6LoWPANs) [29] provides a possible protocol framework to be used with the proposed authentication mechanism. The advantage of applying NDP is to be independent of the data link layer technology and thus use of link-local IPv6 addresses as device identifiers IDs. The role of the HAN gateway could possible be implemented in the border router. To support the host-initiated interactions that allow for sleeping devices, the proposed authentication scheme could leverage on the new address registration mechanism of [29] which has been added to remove the need for routers to use multicast Neighbor Solicitation (NS) to find hosts and to support sleeping devices. If the TA cannot authorize the host an address registration error message can be used to deny access for the host. The host chooses a lifetime of the registration and repeats the Address Registration Option (ARO) periodically (before the lifetime runs out) to maintain the registration. This lifetime can conveniently be set identical to the time for refreshing of session keys. The NDP protocol specification opens for the possibility is to carry the keying material between devices as options in the Neighbor Advertisement (NA) messages [24]. This, however, puts an upper limit to the key size to be used during bootstrapping of 312 bits when using IPv6 over IEEE 802.15.4 networks. Fortunately, with ECC this seems to be a feasible approach for the immediate future.

IX. CONCLUSIONS

The Home Area Network (HAN) plays an essential role in the smart grid by connecting sensors and actuators at the customer premises with the power grid in a secure and dependable manner. The work presented in this paper proposes a secure authentication method for resource-constrained devices in a HAN. The method is based on a pairing-based cryptography that utilizes elliptic curves. A bootstrapping protocol that uses the identity of devices exchanged by a neighbor discovery protocol mechanism is proposed to establish key-pairs between communicating devices. Critical elements of the proposed protocol are validated in a set of prototype implementations. The device authentications method rely on the transistive trust principle. Trust to a new HAN device is given to the HAN gateway by the user, who desires to add a new device to the HAN, and trust is further tranferred from the gateway to end-devices in the HAN.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework program (FP7/2007-2013) under grant agreement n° 317761 (SmartHG).

REFERENCES

[1] V. C. Gungor, B. Lu, and G. P. Hancke, "Opportunities and Challenges of Wireless Sensor Networks in Smart Grid," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557–3564, 2010.

[2] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2004, pp. 71–80.

[3] A. Jøsang, E. Gray, and M. Kinateder, "Simplification and Analysis of Transitive Trust Networks," *Web Intelligence and Agent Systems*, vol. 4, no. 2, pp. 139–161, 2006.

[4] A. He and B. Sarikaya, "IoT Security Bootstrapping: Survey and Design Considerations," pp. 1–17, 9 March, 2015 2015, internet Society. [Online]. Available: https://tools.ietf.org/html/draft-he-6lo-analysis-iot-sbootstrapping-00

[5] H. Li, Z. Jia, and X. Xue, "Application and Analysis of ZigBee Security Services Specification," in *2010 Second International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, vol. 2, 2010, pp. 494–497.

[6] "ZigBee Specification (version 2)," ZigBee Alliance, Tech. Rep. Document 053474r17, January 17 2008.

[7] B. Stelte and G. D. Rodosek, "Thwarting Attacks on ZigBee - Removal of the KillerBee Stinger," in *2013 9th International Conference onNetwork and Service Management (CNSM)*, 2013, pp. 219–226.

[8] X. Boyen, "A Promenade through the New Cryptography of Bilinear Pairings," in *IEEE Information Theory Workshop*, 2006, pp. 19–23.

[9] L. B. Oliveira, D. F. Aranha, C. P. L. Gouvêa, M. Scott, D. F. Câmara, J. López, and R. Dahab, "Tinypbc: Pairings for authenticated identity-based non-interactive key distribution in sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 485–493, March 2011.

[10] D. Boneh and M. Franklin, Eds., *Identity-Based Encryption from the Weil Pairing*, ser. Advances in Cryptology CRYPTO 2001. Springer Berlin Heidelberg, 2001, vol. 2139. [Online]. Available: http://dx.doi.org/10.1007/3-540-44647-8_13

[11] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO 84 on Advances in cryptology*. New York, NY, USA: Springer-Verlag New York, Inc, 1985, pp. 47–53.

[12] M. Sethi, J. Arkko, and A. Keranen, "End-to-end security for sleepy smart object networks," in *2012 IEEE 37th Conference on Local Computer Networks Workshops (LCN Workshops)*, 2012, pp. 964–972.

[13] K.-W. Park, S. S. Lim, and K.-H. Park, "Computationally Efficient PKI-Based Single Sign-On Protocol, PKASSO for Mobile Devices," *IEEE Transactions on Computers*, vol. 57, no. 6, pp. 821–834, 2008.

[14] V. S. Miller, "Use of elliptic curves in cryptography," in *Lecture Notes in Computer Sciences; 218 on Advances in Cryptology—CRYPTO 85*. New York, NY, USA: Springer-Verlag New York, Inc, 1986, pp. 417–426. [Online]. Available: http://dl.acm.org/citation.cfm?id=18262.25413

[15] E. Barker, W. Barker, W. Burr, W. Polk, M. Smid, P. D. Gallagher, and U. S. For, "NIST Special Publication 800-57 Recommendation for Key Management Part 1: General," 2012.

[16] D. J. Malan, M. Welsh, and M. D. Smith, "Implementing Public-Key Infrastructure for Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 4, pp. 22:1–22:23, sep 2008.

[17] A. Liu and P. Ning, "TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks," in *International Conference on Information Processing in Sensor Networks*, 2008, pp. 245–256.

[18] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, *TinyOS: An Operating System for Sensor Networks*, ser. Ambient Intelligence. Springer Berlin Heidelberg, 2005, pp. 115–148.

[19] L. B. Oliveira, M. Scott, J. Lopez, and R. Dahab, "TinyPBC: Pairings for Authenticated Identity-Based Non-Interactive Key Distribution in Sensor Networks," in *5th International Conference on Networked Sensing Systems (INSS 2008)*, 2008, pp. 173–180.

[20] H. Nicanfar, P. Jokar, and V. C. M. Leung, "Efficient Authentication and Key Management for the Home Area Network," in *2012 IEEE International Conference on Communications*, 2012, pp. 878–882.

[21] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo, "Secure Key Issuing in ID-Based Cryptography," in *Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, ser. ACSW Frontiers '04, vol. 32, Darlinghurst, Australia, Australia, 2004, pp. 69–74.

[22] T. S. Hjorth and R. Torbensen, "Trusted Domain: A Security Platform for Home Automation," *Computers & Security*, vol. 31, no. 8, pp. 940–955, 11 2012.

[23] M. Wang, H. Hu, and G. Dai, "An Efficient Signature Scheme Based on Tate Pairing," in *Second International Conference on Innovative Computing, Information and Control*, 2007, pp. 616–620.

[24] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," sep 2007, updated by RFC 5942. [Online]. Available: http://www.ietf.org/rfc/rfc4861.txt,

[25] B. L. Titzer, D. K. Lee, and J. Palsberg, "Avrora: scalable sensor network simulation with precise timing," in *Proceedings of the 4th international symposium on Information processing in sensor networks*, ser. IPSN '05. Piscataway, NJ, USA: IEEE Press, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1147685.1147768

[26] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," p. 305, 2006, iEEE Std 802.15.4-2006.

[27] MEMSIC, "Iris data sheet," available at: http://www.memsic.com/userfiles/files/datasheets/wsn/iris_datasheet.pdf

[28] E. B. Barker and A. L. Roginsky, "SP 800-131A. Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths," National Institute of Standards & Technology, Tech. Rep., 2011.

[29] Z. Shelby, S. Chakrabarti, E. Nordmark, and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)," nov 2012, internet Society, RFC 6775. [Online]. Available: http://www.ietf.org/rfc/rfc6775.txt