Low-Cost Standard Public Key Cryptography Services for Wireless IoT Systems

Muslum Ozgur Ozmen
Oregon State University
Corvallis, Oregon, USA
ozmenmu@oregonstate.edu

Attila A. Yavuz
Oregon State University
Corvallis, Oregon, USA
attila.yavuz@oregonstate.edu

ABSTRACT

Internet of Things (IoT) is an integral part of application domains such as smart-home and digital healthcare. Various standard public key cryptography techniques (e.g., key exchange, public key encryption, signature) are available to provide fundamental security services for IoTs. However, despite their pervasiveness and well-proven security, they also have been shown to be highly energy costly for embedded devices. Hence, it is a critical task to improve the energy efficiency of standard cryptographic services, while preserving their desirable properties simultaneously.

In this paper, we exploit synergies among various cryptographic primitives with algorithmic optimizations to substantially reduce the energy consumption of standard cryptographic techniques on embedded devices. Our contributions are: (i) We harness special precomputation techniques, which have not been considered for some important cryptographic standards to boost the performance of key exchange, integrated encryption, and hybrid constructions. (ii) We provide self-certification for these techniques to push their performance to the edge. (iii) We implemented our techniques and their counterparts on 8-bit AVR ATmega 2560 and evaluated their performance. We used microECC library and made the implementations on NIST-recommended secp192 curve, due to its standardization. Our experiments confirmed significant improvements on the battery life (up to 7×) while preserving the desirable properties of standard techniques. Moreover, to the best of our knowledge, we provide the first open-source framework including such set of optimizations on low-end devices.

Keywords: Internet of Things; Cryptographic Optimizations; Efficient Implementations; Wireless Network Security.

1 INTRODUCTION

Internet of Things (IoT) is a heterogeneous system comprised of interrelated smart-objects and sensors. Due to IoTs' pervasiveness and impact on the real-life applications, it is critical to guarantee their security. Especially, fundamental security services such as authentication, integrity, and confidentiality are required for any viable IoT.

Although various standard cryptographic techniques exist ([2, 3, 19]), the vast majority of them may not fully meet the needs of IoTs, especially when such systems involve resource-limited devices. In particular, despite the recent progress on the capabilities of off-the-shelf embedded systems (e.g, AVR ATmega 2560), the energy-constraints of such devices still pose a critical limitation.

Below, we first discuss the limitations of some alternatives and specify the research gap to be addressed. We then present our contribution by summarizing the desirable properties of our schemes.

Problem Statement and Research Gap: Symmetric primitives are preferred for resource-limited devices due to their computational efficiency, however, Public Key Cryptography (PKC) is also an essential tool for IoTs: (i) Energy efficient PKC is necessary for the management/distribution of symmetric keys in ubiquitous IoT systems. (ii) Symmetric primitives might not be scalable for large-distributed systems [18], while PKC can achieve scalability for large systems. (iii) Symmetric primitives do not offer public verifiability and non-repudiation, which are highly desirable for some IoT applications such as payment systems, secure audit logging, and digital forensics (medical devices). On the other hand, to pervasively deploy PKC in resource-limited IoT systems, the efficiency of PKC primitives should be substantially improved and optimized.

Many techniques are proposed to improve the efficiency of PKC [2, 3]. Improved standards include key exchange (HMQV [12]), integrated encryption (ECIES [14]) and hybrid constructions (Signcryption [19]). To further improve these techniques, lightweight signatures [9], self-certified key exchange [11], and efficient Elliptic Curve (EC) variants [4, 8] have been introduced. Despite their merits, there is a research gap that prevents the full utilization of performance benefits of these techniques for IoT systems:

(i) The integrated schemes and self-certified constructions have various common operations to be synergized. Yet, these primitives are considered in isolation. (ii) These common operations have the potential to receive significant benefits from special algorithmic optimizations [6], which have not been explored for integrated and self-certified cryptographic techniques. (iii) A comprehensive energy consumption analysis of such improved cryptographic techniques on modern embedded devices are currently missing in the literature. (iv) An open-source framework of energy efficient schemes, specifically for IoT applications for public use is necessary.

Our Contribution: Towards filling the aforementioned research gaps, we propose a series of cryptographic optimizations that exploit synergies and algorithmic techniques to enable high efficiency and minimum energy consumption for wireless IoT systems.

- *Improving Battery life with Low Storage Overhead*: One of the costly operations in standard PKC suites is EC scalar multiplication (*Emul*). We observe that it can be significantly accelerated with

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IoTS&P'17, November 3, 2017, Dallas, TX, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5396-0/17/11…$15.00
https://doi.org/10.1145/3139937.3139940

Boyko-Peinado-Venkatesan (BPV) technique [6], whose potential is not investigated for major cryptographic suites (ECHMQV [12], ECIES [14] and Signcryption [19]). We provide, to the best of our knowledge, the first realization of BPV for these suites on embedded devices. We also present further optimizations that we refer to as Designated BPV (DBPV). Our improved suites achieve significantly lower energy consumption with a small constant-storage overhead. Note that the traditional precomputation techniques incur linear token storage/re-generation costs (a token per-item), which are not feasible for memory limited IoT devices. Moreover, it is shown in [15] that the re-generation of tokens may require more energy and time than just following the standard protocol.

• *Eliminating Certification Overhead*: In aforementioned cryptographic suites, the sender creates an ephemeral ECDH key to be incorporated in encryption and/or signatures. We notice that by transforming this step into a self-certified ECDH operation, for instance via Arazi-Qi (AQ) [2], it is possible to seamlessly eliminate the verification/transmission overhead introduced by certificates.

• *Integration of Optimizations to Standard Suites*: We identify that self-certification synergizes well with BPV, providing further efficiency gain. Our analysis shows significant performance gains for both fixed key exchange and integrated protocols. With these improvements: (i) Our proposed scheme AQ-BPV achieves almost 3× faster key exchange than ECDH with ECDSA certificate, where the transmission cost of the certificate is also eliminated (see Table 2). (ii) Our improved schemes with AQ, BPV and DBPV eliminate the overhead of certificates and improve execution time by up to 7× (see Table 3) for integrated schemes such as ECIES and Signcryption.

• *Experimental Evaluation and Open-Source Framework*: We implemented our techniques and their counterparts on an 8-bit ATmega 2560 microcontroller which is widely used in IoT applications due to its flexibility and low-power consumption [16, 17]. Our experiments confirmed that our schemes achieve approximately 7× improvement in terms of battery life and computation time (see Section 4). Moreover, to the best of our knowledge, there is no open-source library for these cryptographic suites and the improvements we have adopted to low-end embedded devices. Therefore, we are putting an effort for the adoption of our optimizations and these cryptographic suites by making our implementations open-source[1].

Limitations: BPV introduces the storage of a 11.25 KB (constant-size) table, and when DBPV is also utilized, this storage overhead increases to 18.75KB. However, we show that such storage is feasible even to 8-bit devices like ATmega 2560 microcontroller, and provides up to 7× time and energy efficiency. Therefore, we believe it is a useful trade-off. The limitation of AQ protocol (which provides self-certification) is that a key generation center (KGC) needs to calculate and distribute the keys to the nodes. While this approach is certainly feasible to be employed in certain IoT applications (e.g., smart airport/city systems), it may not be for some other applications. As self-certification removes all certification overhead, we believe it is useful to adopt AQ protocol when it is feasible.

Note that our optimizations are not tightly coupled. Therefore, for the applications that are not suitable for AQ protocol, BPV and DBPV still provide significant improvements (vice versa). Moreover, these improvements are achieved by preserving the core operations of the base schemes, so they retain their security properties as well as permitting an easy adoption for real-life applications.

2 PRELIMINARIES

We first outline notation in Table 1, and then describe building blocks used by our proposed schemes as follows:

Table 1: Notation followed to describe schemes.

$\mathbf{G}$	Generator group point
q	Order of group
d	*Private Key* of CA
$\mathbf{D}$	*Public Key* of CA where $\mathbf{D} = d \times \mathbf{G}$
x_i	Fixed *Private Key* of Node i
$\mathbf{U}_i$	Fixed *Public Key* of Node i
ID_i	Identification of Node i
m	Message
×	Elliptic Curve Scalar Multiplication

Elliptic Curve (EC) points are shown in **bold**.

Arazi-Qi (AQ) Self-Certified Ephemeral Scheme: Arazi-Qi (AQ) [2] proposed a simple yet efficient self-certified ECDH scheme. During the offline phase, all participants in the system are given a self-certified ECDH private/public key pair by a CA. At the online phase, any two entities with valid self-certified key pair can establish a symmetric key without requiring the transmission and verification of ECDH certificates. In Figure 1, we outline an ephemeral AQ variant proposed by Hang et. al. in [11], which offers higher security guarantees.

AQ-Hang.Offline (offline calculations performed by CA)

```
1: $b_a \xleftarrow{\$} \mathbf{Z}_q, \mathbf{U}_a \leftarrow b_a \times \mathbf{G}$.
2: $x_a \leftarrow [H(ID_a, \mathbf{U}_a) \cdot b_a + d]$ and repeat 1-2 for node B.
3: Node A $\leftarrow (x_a, \mathbf{U}_a)$, Node B $\leftarrow (x_b, \mathbf{U}_b)$.
```

AQ-Hang.Online (online calculations)

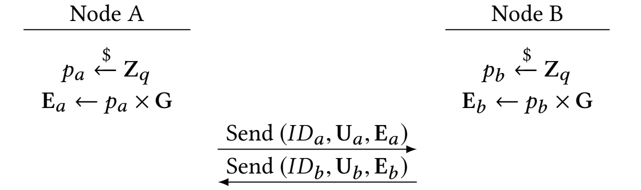

Node A
Node B
$p_a \xleftarrow{\$} \mathbf{Z}_q$
$\mathbf{E}_a \leftarrow p_a \times \mathbf{G}$
$p_b \xleftarrow{\$} \mathbf{Z}_q$
$\mathbf{E}_b \leftarrow p_b \times \mathbf{G}$
Send $(ID_a, \mathbf{U}_a, \mathbf{E}_a)$
Send $(ID_b, \mathbf{U}_b, \mathbf{E}_b)$

Figure 1: Ephemeral AQ variant by Hang et. al. [11]

Node A: $\mathbf{K}_{ab} = x_a \times [H(ID_b||\mathbf{U}_b) \times \mathbf{U}_b + \mathbf{D}] + p_a \times \mathbf{E}_b$.
Node B: $\mathbf{K}_{ab} = x_b \times [H(ID_a||\mathbf{U}_a) \times \mathbf{U}_a + \mathbf{D}] + p_b \times \mathbf{E}_a$.

As $x_a \times [H(ID_b||\mathbf{U}_b) \times \mathbf{U}_b + D] = x_b \times [H(ID_a||\mathbf{U}_a) \times \mathbf{U}_a + \mathbf{D}] = x_a \cdot x_b \times \mathbf{G}$ is constant for both nodes (which is the fixed key in AQ [2]), they can store this value and use it in future key exchanges. In the online phase, there are only two *Emul* for each node. This also decreases the bandwidth as U_a and U_b are transferred only once.

Boyko-Peinado-Venkatesan (BPV) Technique [6]: This technique reduces the computational cost of a full scalar multiplication to only a few EC additions with the expense of a small-constant size table storage.

$\Gamma \leftarrow BPV.Offline(n)$, n : Number of precomputed pairs.

```
1: $p_i \xleftarrow{\$} \mathbf{Z}_q, \mathbf{P}_i \leftarrow p_i \times \mathbf{G}$, and store pairs $\Gamma = \langle(p_i, \mathbf{P}_i)\rangle_{i=1}^{n}$
```

$(r, \mathbf{R}) \leftarrow BPV.Online(\Gamma)$

```
1: Generate a random set $S \subset [1, n]$, where $|S| = k$.
2: $r \leftarrow \sum_{i \in S} p_i$ and $\mathbf{R} = r \times \mathbf{G} = \sum_{i \in S} \mathbf{P}_i$.
```

[1] https://github.com/ozgurozmen/OptimizedPKCSuite

3 PROPOSED TECHNIQUES

Our target suites are key exchanges (ECHMQV [12], AQ [2]), and integrated protocols (ECIES [14] Signcryption [19]). Our rationale for selecting these cryptographic suites can be summarized as follows: (i) Although ECDH with certificates is very common in practice (SSL/TLS), it is very costly for IoT systems. Therefore, we improve AQ scheme, a lightweight self-certified key exchange protocol, and ECHMQV scheme as it was standardized in IEEE P1363 [1]. (ii) Integrated schemes provide both authentication and encryption with a less cost than considering these two apart. Selected integrated schemes are also standardized (ECIES - IEEE P1363 [1], Signcryption - ISO/IEC 29150:2011) and extensively used in practice.

• *Seamless Integration of Self-Certification:* These cryptographic techniques require a certificate to be transmitted and verified to ensure the authenticity of the public key(s). We notice that these techniques generate an Elgamal encryption key as an (ephemeral) ECDH key. This key is directly used in ECHMQV and ECIES, and also incorporated into joint signature/encryption in Signcryption. We exploit this common step to enable a self-certification by adopting AQ protocol [2]. This strategy permits us to avoid the transmission and verification of certificates but requires all nodes to receive their key set from CA as required by AQ protocol.

• *Constant Size Pre-computation:* Traditional precomputation techniques store a set of pairs $\langle r_i, r_i \times \mathbf{G} \rangle_{i=1}^{N}$ to avoid online scalar multiplications, which incurs a linear memory overhead. Moreover, once these tokens are depleted, the device must re-generate them, which is highly costly [15]. Hence, these techniques are not suitable for battery-limited IoT devices. We observe that BPV (see Section 2) has been overlooked for various standard cryptographic suites. We harness BPV to speed-up operations involving a scalar multiplication with randomness in these cryptographic techniques.

• *Enabling BPV for Designated Public Keys:* Some of these integrated cryptographic techniques require an online scalar multiplication over a public key in the form of $\langle r, r \times \mathbf{U} \rangle$, which cannot be directly speed-up via BPV. However, we observe that it is possible to extend BPV to this setting, if the sender can store a table for each receiver public key $\langle \Gamma_i, \mathbf{U}_i \rangle_{i=1}^{r'}$. In many IoT applications, the number of receivers that an IoT device reports to is generally limited (one or at most a few cloud servers). Hence, we propose to apply BPV to this set of designated public keys, and we refer this strategy to as *Designated BPV (DBPV)*. Please note that *DBPV* might not be applicable if the number of receivers is large for the IoT device.

• *Preserving Security Features of Primitives due to Direct Integration:* All the improved proposed schemes perpetuate security properties of underlying primitives as optimization techniques are integrated directly, without any modification. Therefore, there is no need for separate security proofs of the proposed schemes. Thus, our optimizations can be integrated easily to the existing schemes.

• Improving AQ and ECHMQV Key Exchange:

Scheme I - Ephemeral AQ-BPV: Figure 1 depicts that E_a and E_b are calculated by EC scalar multiplications. Instead, we leverage BPV to minimize this overhead. Thus, in the offline phase, precomputation steps of BPV are followed by both parties so that in the online phase E_a and E_b are calculated only with EC additions.

Scheme II - ECHMQV with AQ-BPV: ECHMQV protocol needs a prior ECDH key exchange, which requires certified public keys [12]. Instead, we make ECHMQV self-certified by adopting Fixed AQ protocol. Prior to online calculations, nodes A and B follow Fixed AQ protocol [2]. Thus, private and public key pair of nodes are $x_a = [H(ID_a, \mathbf{U_a}) \cdot b_a + d]$ where $\mathbf{U_a} = b_a \times \mathbf{G}$ and $x_b = [H(ID_b, \mathbf{U_b}) \cdot b_b + d]$, where $\mathbf{U_b} = b_b \times \mathbf{G}$. Furthermore, ECHMQV also receives benefits from BPV, especially in deriving ephemeral session keys.

```
Node A: $(p_a, \mathbf{P}_a) \leftarrow BPV.Online(\Gamma_a)$
Node B: $(p_b, \mathbf{P}_b) \leftarrow BPV.Online(\Gamma_b)$
Node A and Node B exchange $\mathbf{P}_a$ and $\mathbf{P}_b$
$e_1 \leftarrow H(\mathbf{P}_a || \mathbf{U_b})$ and $e_2 = H(\mathbf{P}_b || \mathbf{U_a})$
$\sigma_A \leftarrow (p_a + e_1 \cdot x_a) \times (\mathbf{P}_b + e_2 \times H(ID_b || \mathbf{U_b}) \times \mathbf{U_b} + \mathbf{D})$
$\sigma_B \leftarrow (p_b + e_2 \cdot x_b) \times (\mathbf{P}_a + e_1 \times H(ID_a || \mathbf{U_a}) \times \mathbf{U_a} + \mathbf{D})$
$\mathbf{K}_{ab} = H(\sigma_A) = H(\sigma_B)$
```

Note that the values $H(ID_b || \mathbf{U_b}) \times \mathbf{U_b} + \mathbf{D}$ and $(H(ID_a || \mathbf{U_a}) \times \mathbf{U_a} + \mathbf{D})$ can be calculated only once, prior to online communications. With all these optimizations combined, a total of four EC scalar multiplications can be reduced to $(3 + 2k)$ *Eadd* (*Eadd* denotes EC additions, $k = 8$), which offers a significant performance gain.

• Improving Integrated Schemes:

Scheme III - ECIES with AQ-BPV: As in ECHMQV, we first integrate Fixed AQ into ECIES to achieve self-certified fixed ECDH keys. Therefore, $x_a = [H(ID_a, \mathbf{U_a}) \cdot b_a + d]$ and $x_b = [H(ID_b, \mathbf{U_b}) \cdot b_b + d]$, where $\mathbf{U_a} = b_a \times \mathbf{G}$ and $\mathbf{U_b} = b_b \times \mathbf{G}$. Moreover, the sender uses BPV to eliminate an online EC scalar multiplication. Finally, $H(ID_b || \mathbf{U_b}) \times \mathbf{U_b}$ is calculated only once at the offline phase.

Sender

```
1: $(p_a, \mathbf{P}_a) \leftarrow BPV.Online(\Gamma_a)$
2: $\mathbf{Z} \leftarrow p_a \times [H(ID_b || \mathbf{U_b}) \times \mathbf{U_b} + \mathbf{D}]$, where $\mathbf{Z} = (x_1, y_1)$
3: $k_e || k_m \leftarrow KDF(S || S_1)$, where $S_1$ is public (e.g., $ID_a$) and $S = x_1$
4: $c \leftarrow \mathcal{E}_{k_e}(m)$, $d \leftarrow MAC_{k_m}(c || S_2)$, where $S_2$ is public (e.g., $ID_b$)
5: Send $(\mathbf{P}_a, c, d)$ to the receiver
```

Receiver

```
1: $\mathbf{Z} \leftarrow x_b \times \mathbf{P}_a$, where $\mathbf{Z} = (x_1, y_1)$
2: $k_e || k_m \leftarrow KDF(S || S_1)$, where $S = x_1$
3: If $d = MAC_{k_m}(c || S_2)$ then $m \leftarrow \mathcal{D}_{k_e}(c)$
```

ECIES can be further improved with DBPV as follows:

Scheme IV - ECIES with AQ-DBPV: In addition to computing $\mathbf{P}_a$ with BPV (Sender Step 1), we observe that the values for public key $\mathbf{Z}$ can also be precomputed and stored in a similar way. That is, our precomputation table also includes values for $(p_a, \mathbf{Z} = p_a \times [H(ID_b || \mathbf{U_b}) \times \mathbf{U_b} + \mathbf{D}])$. When sender needs to generate S, she just uses these precomputed values to obtain Z with only k *Eadd* operations. Therefore, we denote these DBPV operations as $(p_a, \mathbf{P}_a, \mathbf{Z}) \leftarrow DBPV.Online(\Gamma_a)$. Notice that, after these improvements, there is no EC scalar multiplications but only $2k$ *Eadd* operations at the sender side.

Scheme V - Signcryption with AQ-DBPV: We notice that Signcryption is initiated by sender performing an *Emul* over the public key of the receiver (a DH key in base Signcryption [19]). This implies that Signcryption can also benefit from both AQ and DBPV optimizations. That is, we first make Signcryption self-certified, where the nodes follow fixed AQ protocol prior to online communication as, $x_a = [H(ID_a, \mathbf{U_a}) \cdot b_a + d]$ and $x_b = [H(ID_b, \mathbf{U_b}) \cdot b_b + d]$, where $\mathbf{U_a} = b_a \times \mathbf{G}$ and $\mathbf{U_b} = b_b \times \mathbf{G}$, respectively. Furthermore, as in ECIES, the sender performs $(p_a, \mathbf{P}_a, \mathbf{Z}) \leftarrow DBPV.Online(\Gamma_a)$.

Table 2: Performance of existing and improved schemes on 8-bit ATmega 2560.

Protocol[¶]	CPU cycles	CPU Time[†](s)	Code Size (Byte)	Bandwidth (Byte)	Cert. Overhead
ECDH+ECDSA+Certificate	51 842 165	3.24	34698	72	yes
AQ	33 638 127	2.10	33192	**24**	**no**
ECHMQV+Certificate	68 961 784	4.31	35788	72	yes
Our Proposed Improved Schemes with Optimization					
AQ-BPV	**19 040 364**	**1.19**	45712	**24**	**no**
ECHMQV with BPV	**55 204 982**	**3.45**	45872	72	yes
ECHMQV with AQ-BPV	**36 164 203**	**2.26**	45872	**24**	**no**

¶ All protocols are implemented as ephemeral key exchange schemes. All comparisons are made for the online phases of these schemes.
† CPU times are based on the first online phase of the protocols. After the first phase, where the public key should be verified, verification cost (1.19s) is removed, until public keys are renewed.

Table 3: Performance of existing and improved schemes on 8-bit ATmega 2560.

Protocol	CPU cycles	CPU Time[†](s)	Code Size (Byte)	Bandwidth (Byte)	Cert. Overhead
ECIES with ECDSA+Certificate	52 007 520	3.25	34876	96	yes
Signcryption with ECDSA+Certificate	39 680 214	2.48	36418	96	yes
Our Proposed Improved Schemes with Optimization					
ECIES with BPV	**38 082 365**	**2.38**	48274	96	yes
ECIES with DBPV	**24 163 017**	**1.51**	55004	96	yes
Signcryption with DBPV	**22 563 256**	**1.41**	49318	96	yes
ECIES with AQ-BPV	**19 040 148**	**1.19**	48274	**48**	**no**
ECIES with AQ-DBPV	**5 122 403**	**0.32**	55004	**48**	**no**
Signcryption with AQ-DBPV	**3 520 069**	**0.22**	49318	**48**	**no**

† CPU times are based on the first online phase of the protocols. After the verification of the certificate, the verification cost (1.19s) is removed, until public keys are renewed.

```
Sender
  $(p_a, \mathbf{P}_a, \mathbf{Z}) \leftarrow DBPV.Online(\Gamma_a)$, where $\mathbf{Z} = (x_1, y_1)$
  $k_e || k_m \leftarrow H(x_1)$
  $r \leftarrow H(k_m || m)$ and $s \leftarrow p_a \cdot (r + x_a)^{-1} \bmod q$
  $c \leftarrow \mathcal{E}_{k_e}(m)$, output $(c, r, s)$
Receiver
  $\mathbf{Z} = (s \cdot x_b) \times [(H(ID_a || \mathbf{U_a}) \times \mathbf{U_a} + \mathbf{D}) + r \times \mathbf{G}]$
  $k_e || k_m \leftarrow H(x_1)$, where $\mathbf{Z} = (x_1, y_1)$
  $m \leftarrow \mathcal{D}_{k_e}(c)$, accepted if $H(k_m || m) = r$
```

• **Security of Proposed Schemes:** Our security depends on two well-known primitives, AQ and BPV (considering DBPV is just an extension of BPV and incorporates its security).

The security of BPV is well-analyzed and relies on the hardness of Hidden Subset Sum Problem [6]. Moreover, the security of BPV with an integration to Elliptic Curve Discrete Logarithm Problem (ECDLP) based protocols (e.g., ECDSA) has been investigated in [3]. Specifically, the BPV with ECDLP based signatures rely on Affine Hidden Subset Sum Problem. Given that our adoption of BPV into ECDLP-based key exchange protocols, integrated scheme, and Signcryption adhere these principles, our techniques preserve these security guarantees.

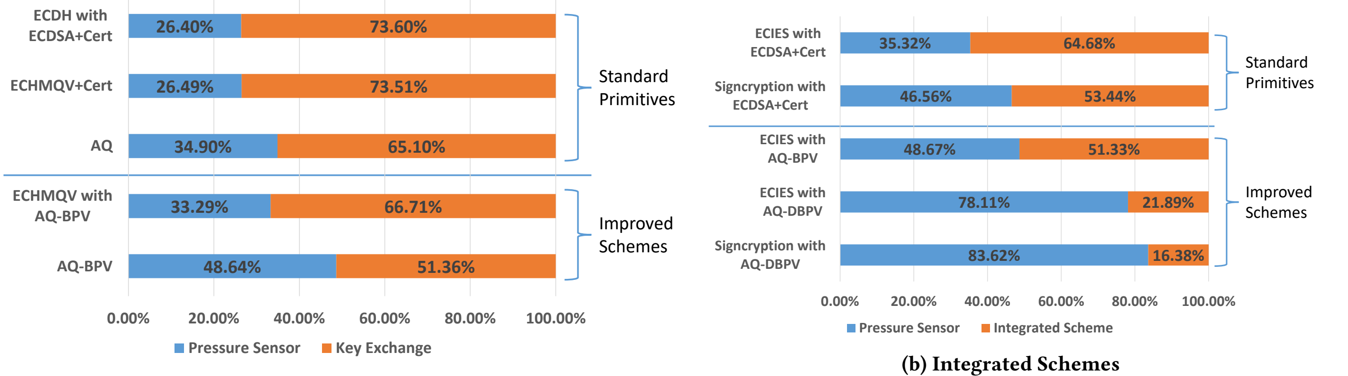
Rest is to show that self-certification does not impact the security of the proposed schemes. As stated by Bernstein in [5], the signature $s = y - H(m||R) \cdot r \bmod q$ in *Schnorr* is a linear combination of the permanent private key y and the ephemeral private key r, with coefficients 1 and $H(m||R)$, respectively. Therefore, it is possible to modify these coefficients by any function of m and R, which yields several variants of *Schnorr* signature. Such variants are also called as “Schnorr-like signatures" as discussed in [5, 10]. Although it is not discussed in the original AQ paper [2], it is depicted in Figure 1 that the private key assigned to nodes is in this form. Basically, in AQ scheme, the private keys are Schnorr-like signatures that are generated by the certification authority in off-line phase and are verified during the key establishment phase. Hence, the security of AQ scheme relies on the security of Schnorr-like signatures, which is well-analyzed.

4 PERFORMANCE EVALUATION

Experimental Setup and Evaluation Metrics: We implemented our schemes and their counterparts on an 8-bit ATmega 2560 microcontroller. ATmega 2560 is a very lightweight device and used commonly in practice for IoT applications, especially in medical devices [16, 17], where there are critical time and energy constraints. AVR ATmega 2560 is an 8-bit microcontroller with 256 KB flash memory, 8KB SRAM and 4 KB EEPROM and its maximum clock frequency is 16MHz. During our experiments, ATmega 2560 was powered by a 2200 mAh power pack. This enabled us to use a DC power monitor/ammeter connected between the battery and processor to monitor the current drawn. Moreover, the experimental current results are compared with the datasheet of the processor[2]. All of the schemes are implemented using microECC library [13].

We selected our elliptic curve as the NIST-recommended secp192 [7] (security parameter $\kappa = 96$). Although there are more efficient curves such as Curve25519 [4] and FourQ [8], NIST curves are the most common ones which are deployed in practice due to their standardization. Moreover, Curve25519 and FourQ offer very fast elliptic curve additions, therefore, we believe, our improvements would be even more effective in these curves. However, in this paper

[2] http://www.atmel.com/Images/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf



(a) Key Exchange Protocols

(b) Integrated Schemes

Figure 2: Energy comparison with IoT sensor (pressure) on 8-bit ATmega 2560

we are following the conservative approach which is not in our favor and use the NIST recommended curves to show our techniques can achieve these numbers even in the slower but standardized curves.

Our evaluation metrics include computation, code size (for ATmega 2560), communication, memory overhead, and energy consumption. We measured the energy consumption with the formula $E = V \cdot I \cdot t$, where $V = 5$ Volts (required by ATmega2560), and t is the computation time (based on clock cycles) as in [3].

In our long-term experiments (to monitor the energy consumption), we focused on the dominative costs for all schemes. Therefore, we did not take the effect of certificate verification into consideration, as this will happen in the first online communication and may not be repeated until the receiver renews its public key. However, even if this cost was also considered, our advantages in terms of energy efficiency would increase.

Performance Evaluation and Comparison: Analytical comparison can be found in Appendix A. We give the experimental evaluation and comparison for key exchange and integrated protocols in Tables 2 and 3, respectively. Our experiments confirmed significant improvements in terms of both CPU time and energy consumption. Moreover, besides saving more energy as compared to ECDH with certificates, they are more communication efficient, by reducing 48 Byte communication overhead. Our optimizations offer even better improvements for integrated protocols. ECIES and Signcryption with AQ-DBPV improve their base schemes for CPU time and energy efficiency by 6.44× and 5.86×, respectively.

In Figure 2, we examined how much energy is required for cryptographic operations as compared to a BMP183 Pressure/Altitude Sensor[3] on ATmega 2560. To calculate the energy consumption of BMP183, we checked the datasheet and observed that the current drawn by the sensor is 5μA and it operates at 2.5V. The sampling rate for this sensor is selected as 30 minutes, and the energy consumed by the sensor is calculated with the formula $E = V \cdot I \cdot t$. Additionally, ATmega 2560 consumes energy to read the data and also during the wait time. These energy consumptions are also taken into consideration. Results in Figure 2 show that the cryptographic operations consume up to 73.6% of the battery. With our optimizations, this overhead is decreased to 51.36% and 16.38% for key exchange and integrated schemes, respectively.

Furthermore, we analyzed the time that ATmega 2560 can operate without a battery replacement/charge when both IoT sensor and cryptographic operations are used, and the sampling rate is 30 minutes. This analysis showed how much our optimizations on cryptographic operations affect the overall energy consumption of the IoT application. We used the data presented in Figure 2 to analyze battery replacement time. If ECDH with ECDSA certificate or ECHMQV with ECDSA certificate were used, the battery would be drained in 50 days, this is increased to 92 days with AQ-BPV. Moreover, Signcryption with certificates and ECIES with certificates drain the battery in 88 and 67 days respectively. With our improved schemes, these numbers increase to 158, 148 days.

5 CONCLUSION

Standard cryptographic suites offer high-security guarantees, but their high energy consumption poses an obstacle towards their broad adoption for battery-limited devices, which are an integral part of IoT applications (e.g., smart-home, healthcare). In this paper, we develop a series of algorithmic improvements and optimizations that can be applied to a vast range of cryptographic techniques with only a minimal modification. It is central to our techniques to enable self-certification and small-constant size precomputation capabilities for prominent key exchange, integrated encryption, and hybrid cryptographic constructions. We fully implemented our techniques and provided a comprehensive experimental evaluation of modern embedded systems to assess their practicality for real-life applications. Our experimental analysis confirmed up to 7× battery life improvements over the standard cryptographic techniques by introducing only a small-constant storage overhead. Our improvements adhere the core design properties of their base cryptographic standards, and can also be potentially adopted to other similar cryptographic techniques.

ACKNOWLEDGMENTS

This material is based upon work supported by the NSF CAREER Award CNS-1652389. The authors would also like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] 2004. IEEE Standard Specifications for Public-Key Cryptography - Amendment 1: Additional Techniques. *IEEE Std 1363a-2004 (Amendment to IEEE Std 1363-2000)*

[3] https://cdn-shop.adafruit.com/datasheets/1900_BMP183.pdf

(Sept 2004), 1–167. https://doi.org/10.1109/IEEESTD.2004.94612

[2] O. Arazi and H. Qi. 2005. Self-Certified Group Key Generation for Ad Hoc Clusters in Wireless Sensor Networks. In *Proceedings of IEEE International Conference on Computer Communications and Networks (ICCCN '05)*. 359–364.

[3] G. Ateniese, G. Bianchi, Angelo Capossele, and Chiara Petrioli. 2013. Low-cost Standard Signatures in Wireless Sensor Networks: A Case for Reviving Pre-computation Techniques?. In *Proceedings of NDSS 2013*. San Diego, CA.

[4] Daniel J. Bernstein. 2006. *Curve25519: New Diffie-Hellman Speed Records*. Springer Berlin Heidelberg, 207–228. https://doi.org/10.1007/11745853_14

[5] Daniel J. Bernstein. 2015. Multi-user Schnorr security, revisited. *IACR Cryptology ePrint Archive* 2015 (2015), 996. http://eprint.iacr.org/2015/996

[6] Victor Boyko, Marcus Peinado, and Ramarathnam Venkatesan. 1998. Speeding up discrete log and factoring based schemes via precomputations. In *Advances in Cryptology — EUROCRYPT'98: International Conference on the Theory and Application of Cryptographic Techniques Espoo, Finland, May 31 – June 4, 1998 Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, 221–235.

[7] Certicom Research. 2009. Standards for Efficient Cryptography – SEC 1: Elliptic Curve Cryptography. http://www.secg.org/download/aid-780/sec1-v2.pdf.

[8] Craig Costello and Patrick Longa. 2015. *FourQ: Four-Dimensional Decompositions on a Q-curve over the Mersenne Prime*. Springer Berlin Heidelberg, Berlin, Heidelberg, 214–235. https://doi.org/10.1007/978-3-662-48797-6_10

[9] Benedikt Driessen, Axel Poschmann, and Christof Paar. 2008. Comparison of Innovative Signature Algorithms for WSNs. In *Proceedings of the First ACM Conference on Wireless Network Security (WiSec '08)*. ACM, 30–35.

[10] S. Galbraith, J. Malone-Lee, and N.P. Smart. 2002. Public key signatures in the multi-user setting. *Inform. Process. Lett.* 83, 5 (2002), 263 – 266.

[11] Isabelle Hang, Markus Ullmann, and Christian Wieschebrink. 2011. Short Paper: A New Identity-based DH Key-agreement Protocol for Wireless Sensor Networks Based on the Arazi-Qi Scheme. In *Proceedings of the Fourth ACM Conference on Wireless Network Security (WiSec '11)*. ACM, New York, NY, USA, 139–144.

[12] Hugo Krawczyk. 2005. HMQV: A High-Performance Secure Diffie-Hellman Protocol. Cryptology ePrint Archive, Report 2005/176. (2005). http://eprint.iacr.org/2005/176.

[13] Ken MacKay. 2013. micro-ecc: ECDH and ECDSA for 8-bit, 32-bit, and 64-bit processors. Github Repository. (2013). https://github.com/kmackay/micro-ecc

[14] D. Pointcheval. 2000. PSEC-3: Provably Secure Elliptic Curve Encryption Scheme. (2000).

[15] A. Singla, A. Mudgerikar, I. Papapanagiotou, and A. A. Yavuz. 2015. HAA: Hardware-Accelerated Authentication for internet of things in mission critical vehicular networks. In *MILCOM 2015 - 2015 IEEE Military Communications Conference*. 1298–1304.

[16] P. Szakacs-Simon, S. A. Moraru, and F. Neukart. 2012. Signal conditioning techniques for health monitoring devices. In *2012 35th International Conference on Telecommunications and Signal Processing (TSP)*. 610–614.

[17] P. Szakacs-Simon, S. A. Moraru, and L. Perniu. 2012. Pulse oximeter based monitoring system for people at risk. In *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*. 415–419.

[18] A. A. Yavuz and P. Ning. 2009. BAF: An Efficient Publicly Verifiable Secure Audit Logging Scheme for Distributed Systems. In *Proceedings of 25th Annual Computer Security Applications Conference (ACSAC '09)*. 219–228.

[19] Y. Zheng. 1997. Digital Signcryption or How to Achieve Cost(Signature & Encryption) << Cost(Signature) + Cost(Encryption). In *Proceedings of Advances in Cryptology (CRYPTO '97)*. 165–179.

APPENDIX

A ANALYTICAL ANALYSIS

Analytical comparison of our techniques with their state-of-the-art counterparts are depicted in Table 4. One may notice that the improvements enabled by our two-stage optimizations are: (i) BPV permitted us to reduce the cost of *Emul* operations to *k Eadd* (where $k = 8$ as in [3]), which offers significant performance gains. DBPV further amplified this gain by requiring slightly more storage (only possible with a small receiver set). (ii) The integration of certified ECDH via AQ enabled us to eliminate the transmission and verification of certificates for the initial key exchange operations.

We exemplified the impacts of these improvements over Fixed ECHMQV and ECIES schemes. Fixed ECHMQV required 7*Emul* performed by each node. Integrating AQ to ECHMQV, we eliminated the transmission of the certificate along with 2*Emul* computation required for its verification. With the help of BPV, another 2*Emul* were reduced to 2*k Eadd*. Hence, our improved fixed ECHMQV with AQ-BPV scheme only requires three full 3*Emul* along with *Eadd* operations. Similarly, ECIES takes the advantage of AQ by eliminating 2*Emul*. We also integrated BPV and DBPV to ECIES, where each of them reduces the cost of one *Emul* to *k Eadd* on the sender side. Therefore, no full *Emul* is needed in the online phase of the sender. Moreover, the cost of Signcryption is also minimized at the sender side, where there is no *Emul* but only a few *Eadd*.

Table 4: Analytical performance analysis of our schemes with their counterparts.

Protocol	**Sender**					**Receiver**	
	Private Key†	***Public Key***†	***Tag Size***	***Key exchange***¶	***Enc.+Sign / Enc.+MAC***	***Key exchange***¶	***Dec.+Sign / Dec.+MAC***
ECDSA+ECDH+Cert	$\|q\|$	$\|q\|$	$2\|q\|$	$4Emul + 2H + Eadd + 3Mulq$	-	$4Emul + 2H + Eadd + 3Mulq$	-
AQ	$2\|q\|$	$2\|q\|$	-	$2Emul + Eadd$	-	$2Emul + Eadd$	-
Fixed ECHMQV+Cert	$\|q\|$	$\|q\|$	$2\|q\|$	$7Emul + 5H + 2Eadd + 4Mulq$	-	$7Emul + 5H + 2Eadd + 4Mulq$	-
ECHMQV+Cert	$2\|q\|$	$2\|q\|$	$2\|q\|$	$3Emul + 3H + Eadd + Mulq$	-	$3Emul + 3H + Eadd + Mulq$	-
ECIES with ECDSA+Cert	$\|q\|$	$\|q\|$	$\|H\|$	-	$2Emul + 6H$	-	$Emul + 6H$
Signcryption with ECDSA+Cert	$\|q\|$	$\|q\|$	$\|q\| + \|H\|$	-	$Emul + 2H$	-	$2Emul + 2H + Eadd$
Our Proposed Improved Schemes with Optimization							
AQ-BPV	$\Gamma + 2\|q\|$	$\Gamma + 2\|q\|$	-	$\boldsymbol{Emul + (1 + k)Eadd}$	-	$\boldsymbol{Emul + (1 + k)Eadd}$	-
Fixed ECHMQV with AQ-BPV	$\Gamma + 2\|q\|$	$\Gamma + 2\|q\|$	-	$\boldsymbol{3Emul + 3H + (3 + 2k)Eadd + mulq}$	-	$\boldsymbol{3Emul + 3H + (3 + 2k)Eadd + mulq}$	-
ECHMQV with AQ-BPV	$\Gamma + 2\|q\|$	$\Gamma + 2\|q\|$	-	$\boldsymbol{2Emul + 3H + (2 + k)Eadd + mulq}$	-	$\boldsymbol{2Emul + 3H + (2 + k)Eadd + mulq}$	-
ECIES with AQ-BPV	$\Gamma + \|q\|$	$\Gamma + \|q\|$	$\|H\|$	-	$\boldsymbol{Emul + 6H + kEadd}$	-	$Emul + 6H$
ECIES with AQ-DBPV	$(r' + 1)\Gamma + \|q\|$	$(r' + 1)\Gamma + \|q\|$	$\|H\|$	-	$\boldsymbol{2kEadd + 6H}$	-	$Emul + 6H$
Signcryption with AQ-DBPV	$r'\Gamma + \|q\|$	$r'\Gamma + \|q\|$	$\|q\| + \|H\|$	-	$\boldsymbol{kEadd + 2H}$	-	$2Emul + 2H + Eadd$

¶ In designed variant of integrated protocols (i.e., ECIES, Signcryption), the sender knows receiver's public key and ID beforehand. $Emul$ and $Eadd$ denote the costs of EC scalar multiplication over modulus q and EC addition over modulus q, respectively. H and $Mulq$ denote a cryptographic hash and a modular multiplication over modulus q, respectively. k is the BPV parameter that shows how many precomputed pairs are selected in the online phase. Suggested value for $k = 8$ [3]. r' is the constant number of public keys (servers) that the node will communicate.

† $\Gamma = n \cdot |q|$ where n is the number of precomputed pairs. Parameter sizes for n, q and H are: $n = 160$, $|q| = 192$ bit, $|H| = 256$ bit