Delft University of Technology, Bachelor Seminar of Computer Science and Engineering

A Comparative Study on Signature Schemes for IoT Devices

Dan Dan Berendsen,

Daily Supervisor: Miray Ayşen, Responsible Professor: Dr. Zekeriya Erkin
Cyber security group
Department of Intelligent Systems
Delft University of Technology

Abstract

The use of Internet of things (IoT) devices is on the rise and given their involvement with sensitive data the security for these devices is of greater importance. As these devices are connected with the internet being able to have secure communication is crucial. Some of the major aspects for secure communication are identification and message integrity. These can both be provided by signature schemes. As IoT devices are often constrained devices picking a signature scheme is not trivial and depends among other things on storage size, security level and computation costs. One should thus look at how current available solutions compare and their suitability for IoT. This paper therefore provides a comparison of some signature schemes and presented some such as SCDSA and CLS suitable for IoT. Furthermore this paper points out that current solutions may become obsolete with the development of quantum computing. This paper thus also presents some hash based quantum resistant solutions for use in IoT.

1 Introduction

As of 2020 there are already 8.74 billion Internet of things (IoT) devices in use. Furthermore the number of IoT devices has a predicted growth of tripling within the next 10 years [1]. IoT devices are connected devices in a ecosystem where these devices communicate to other related devices to automate home and industry tasks [2]. Some example are temperature sensors in shipping containers or a light sensor in an automated home. As these devices are used, among other things, in people's daily life, medical situations and transportation they can contain sensitive data. An example to this is a smart home that can be controlled with an app on the phone. The communication packets send between the devices thus contain Personal Identifiable Information. This includes the name, phone number, adress and log in details of the user [3]. A leak of this data means a huge break of privacy, it is therefore an important task for IoT manufacturers to guarantee safety and stability for these IoT devices [4]. Some aspects of security for these IoT devices are: the encrypted storage of collected data, secure communication protocols, genuine IoT device identification, system event logging and cryptographic key and certificate storage [5].

This paper focuses on the aspect of genuine IoT device identification which is integral in secure digital communication. For IoT device identification the use of signature schemes is employed. As is quite logical it is important to know if the party you receive a message is the genuine party they claim to be. Moreover it is key to be able to check the integrity of a message as otherwise you would be prone to threat of message replacement/modification attacks by malicious parties. An example where this is crucial is sending an software update to an IoT device. If a malicious party could send an update or modify a legit update to a device which does not do any authentication, the malicious party could sabotage a device to stop working or could gain access to the data contained within a device. For the two issues regarding identification and message integrity signature schemes form a solution. Signature schemes are algorithms that generate a signature that is send along with the message to confirm the identity of the sender and the integrity of the message. Signature schemes often make use of a private key, the message itself and some public key/parameters. A signature is then made with the private key and the message and this signature can then be authenticated with the known public key/parameters.

The aspects of security for IoT devices each have various solutions available which are based on efficiency in speed but also power usage, hardware area, use cases and costs. This paper mainly focuses on the efficiency, key sizes and security level of the various signature schemes. IoT devices in general do not have as much computational power as normal computer and IoT devices are often build for a specific task with constricting parameters. This means that finding the the optimal solution is paramount as IoT devices often have clear constraints. Thus it is important to have a comparison between the available signature schemes for IoT devices. Furthermore for future new solutions it is principal that these new solutions are actually an improvement over existing ones. As such during the comparison it is key to not only research the differences between solutions, but also mention what is lacking overall in all of them. This is done for identifying in which fields there could be improvements and even perhaps already propose a solution for this. This paper thus provides a comparison on signature schemes for IoT devices, comments on their suitable for IoT devices and highlights parts related to signature schemes that could be improved upon given the current status of employed signature schemes.

This paper is structured as follows, in chapter 2 the methodology is discussed, chapter 3 and 4 will investigate the chosen schemes.
This paper has divided the schemes into two main groups: the pre-quantum group (chapter 3) and the post-quantum group (chapter 4) [6]. The pre-quantum group contains signature schemes that are widely used, but could theoretically be cracked by a large enough quantum computer in a reasonable time. The pre-quantum group includes RSA, DSA, and ECC schemes. The post-quantum group contains signature schemes which are resistant against attacks by a quantum computer. In the post-quantum group mainly schemes that have Hash-Based Signatures (HBS) are investigated. The division into these 2 groups has been made as the pre-quantum schemes are still very much viable and in use, but in the future with the development into quantum computers the need to change to the post-quantum group seems inevitable.
Chapter 5 reflects on the ethical implications for this paper, chapter 6 will have a discussion about the results and chapter 7 will conclude this paper and allude to possible future work.

2 Methodology

This paper includes an extensive literature study into signature schemes for IoT devices. Furthermore this paper will evaluate the schemes and make a comparison between the different signature schemes based on certain criteria. These criteria include but are not limited to signature latency, storage space, verification latency and adversary resistance. From the comparison the strengths and weaknesses will be analyzed. In addition possible improvements based on the weaknesses found will be suggested and analyzed.

3 Pre-Quantum IoT security

3.1 RSA

The RSA scheme [8] is a widely known and used cryptographic scheme which also provides the means of signing a message. This section will touch briefly upon the inner workings of signing a message by the RSA scheme.

There are two parties appropriately named Alice and Bob, Alice chooses 2 large prime numbers p, q and gets the product of these two as n. From this public value e and private value d are determined. The selection and relation of e and d differs between the original scheme and some current implementations. Where the original scheme uses the Euler totient function: e * d = 1 (mod (p - 1) * (q - 1)) where d is chosen and e is computed. Some implementations use Carmichael's totient function: e * d = 1 (mod lcm(p - 1, q - 1)) with lcm meaning the lowest common multiple, which is the lowest number that both p and q can divide into. Some schemes using Carmichael's totient function choose e and compute d. Each way does result in private key pair (d, n) and public key pair (e, n). For signing Alice using Message (M) making signature (S) is done by hashing the message (H(M)) and then using the same formula as decrypting messages, thus using the private key. S = $H(M)^d$ (mod) n. For verification Bob the receiver also calculates the hash of the message (H(M)) and uses the same formula as encryption on the signature, thus using the public key. H' = S^e (mod) n and checks if H' and H(M) are equal.

Important to note is that as RSA has it's security reduction to integer factorization, it relies on the difficulty off integer factorization, for RSA to be secure large prime numbers are to be chosen for the scheme. By the National Institute of Standards and Technology (NIST) the recommended minimum RSA key size is 2048 bits [9]. As of 2020 the largest RSA key size which has been factorized and thus cracked is 829 bits [10]. This means that RSA schemes are forced to have an significantly large enough key size to be secure. RSA key sizes and the associated security levels in bits and computation costs will be discussed in section 3.4.

3.2 Digital Signature Algorithm

Digital Signature Algorithm (DSA) is a variant of the El-Gamal Digital Signature [7] which has it's security reduction to the discrete logarithmic problem (DLP), meaning it's strength can be boiled down (is at least as hard to break) to the discrete logarithmic problem. The basis for DLP is g^a = x mod p where finding a is a computationally hard problem given a large prime p. The security off the DSA is thus dependant on a strong exponent and a large prime, these two have a relation which will be more explored in section 3.4.

3.2.1 DSA

For DSA some public parameters will generated and shared. A one way hash function H() is chosen The values of p: the prime number, q: the prime divisor of p-1, g: with random number z then $g = z^{(p-1)/q}\ mod\ p$ are shared. Furthermore as stated DSA uses $g^a\ mod\ p = x$, here a is a random number from [1 .. q-1]. From this equation, x forms the public key and a is the private key. With these values a signature can be made which consists of 2 components: components r and s. Using a random value k and message m, $r = (g^k\ mod\ p)\ mod\ q$ and $s = (k^{-1}(H(m) + a * r))\ mod\ q$. This gives a signature S = (r, s) which is sent along with the message.
Then for verification first 1 <= r <= p-1 and 1 <= s <= p-1 is checked. Should this hold, then a variable v is calculated and compared to s to actually check the signature. For v first compute $w = s^{-1}\ mod\ q$, $u1 = (w * H(m))\ mod\ q$ and $u2 = r * w\ mod\ q$. Then v is calculated as $v = ((g^{u1} * x^{u2})\ mod\ p)\ mod\ q$ and the signature is verified if v = s.

3.2.2 Shortened Complex Digital Signature Algorithm

Shortened Complex Digital Signature Algorithm (SCDSA) [11] is a lightweight scheme that has been proposed for use in IoT devices. The scheme uses a novel method to decrease the bit sizes and increase the performance. SCDSA uses mostly the same parameters: a large prime p, greatest common divider q and a hash function H() (SHA-1 is proposed). Additionally the the scheme uses an unique random k for each signature. The scheme differs to DSA in that it uses a complex number as the base g. This complex number g is chosen from a field Fq which is chosen when p is chosen. SCDSA still uses g^a = x where x is still the public key and parameter a the chosen private key. Then for the r part $g^k\ mod\ p$ is concatenated with the message m and put into the hash function. $r = H(g^k\ mod\ p\ ||\ m)$. Note that g^k is calculated by a squaring and multiplication algorithm. With r the parameter is calculated $s = k/(r + a)\ mod\ q$ which gives the signature S = (r, s)
For verification r' is calculated and compared to r where the signature will be accepted if r'= r. For this the receiver Bob actually calculates g^k by $g^k = (x * g^r)^s\ mod\ p$. With g^k then the same method to get r in signing is used for r', $r' = H(g^k\ ||\ m)$. The paper for SCDSA also provided an analysis on the computation times between DSA and SCDSA. With the same base number simulation have been run for different values of k. For signing SCDSA outperforms DSA. Below is some data extracted from the graph provided by [11].

Table 1. Computation cost of DSA SCDSA

Value of k	DSA	SCDSA	Difference
	Signing	Signing	
20	350ms	300ms	13%
30	825ms	700ms	15%
40	1500ms	1250ms	17%
	Verification	Verification	
20	300ms	30ms	90%
30	700ms	45ms	93%
40	1250ms	80ms	93%

As seen in table 1. SCDSA performs significantly better than DSA. SCDSA achieves better performance by minimizing the amount of operations needed. For example for verification DSA needs multiple operations for w, u1, u2 and v whereas SCDSA only needs 2 operations for g^k and r'. Meaning r' is calculated more efficiently, but with the same security level as DSA.

3.3 Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is based on a elliptic curve over a finite field which satisfies $y^2 = x^3 + ax + b$. ECC require substantially smaller keys size for a the same level of security compared to the RSA scheme. This also leads to faster computation and smaller storage space needed.

3.3.1 Certificateless signature schemes

It is known that public key cryptography (PKC) also known as traditional certificate-based public key cryptography (TC-PKC) face certificate management problems. The challenges with certificate management are well known and extend beyond the goal of this paper, this paper will not go in-depth into signature schemes that do use certificates. From TC-PKC we got Identity-based Public Key Cryptography (ID-PKC), which removes the need for certificate management and the problems associated with it, however ID-PKC suffers from key-escrow problems. To solve both problems Certificateless Public Key Cryptography (CL-PKC) [12] was proposed. Certificateless Signature (CLS) schemes solve the certificate management and key-escrow problems by having the private key be a combination of some secret information from the user and a partial key generated by a key generating centre (KGC). Furthermore because of the many optimization and novelties for CLS schemes, [13] has been chosen for investigation over ECDSA [14].
CLS schemes have 2 main possible vulnerabilities by which a malicious party can try to attack the scheme, these 2 vulnerabilities are defined as adversary types I and II. "A Type I adversary acts as an outsider who can replace the public keys but cannot access the master secret key, whereas a Type II adversary acts as the KGC that can access the master secret key but cannot replace the public keys." [15] This model has been further extended by dividing each adversary into 3 kinds: normal, strong, super adversary. These levels are ordered on their respective attack power with super having the most attack power. [16]
In the paper 'A new provably secure certificateless signature scheme for Internet of Things' [13] an evaluation of past CLS schemes has been provided. The evaluated papers form a compact timeline in the development off CLS schemes. With Al-Riyami and Paterson's [12] being the first CLS scheme proposed. Zhang-Wong-Xu-Feng's [17] first moving CLS schemes into the random oracle model. Choi-Park-Lee's [18] is a novel short CLS scheme which is however prone to Type 1 strong adversaries. Chen-Tso-Horng, et al. [19] is a short CLS scheme which is secure against both adversary types, but has the drawback of higher computational cost. As bilineair pairing calculations, which were so far present in all schemes, are quite inefficient He-Chen-Zhang's [26] came up with the first scheme without bilinear pairings thus increasing the efficiency of the scheme. Gong and Li's [21], Wang-Chen-Long-Mao's[22], Yeh-Su-Choo's [23], Jia-He-Liu-Kim's [24] and Karati-Islam-Biswas [25] are all schemes without bilinear pairings but each having their own drawbacks.

Legend for table 2. (costs from [26] using [27])

Symbol	Operation	actual value	Security	Types covered
Tpr	A bilinear pairing computation	4.2110	insecure	Insecure: all
Tpsm	A pairing-based scalar multiplication	1.7090	normal	Secure: normal
Tppa	A pairing-based point addition	0.0071	strong	Secure: normal, strong
Tsm	A scalar multiplication on elliptic curve	0.4420	super	Secure: all
Tpa	A point addition on elliptic curve	0.0018		
Tmtp	A map-to-point hash function	4.4060		

Table 2. Security performance comparison among different CLS schemes [13].

Scheme	Sign cost	verify cost	Adversary Type I	Adversary Type II
Al-Riyami and Paterson's [12]	3Tpsm + 1Tpr	4Tpr + 1Tpsm	insecure	insecure
Zhang-Wong-Xu-Feng's [17]	3Tpsm	4Tpr	super	super
Choi-Park-Lee's [18]	3Tpsm	3Tpr + 2Tpsm	normal	super
Chen-Tso-Horng, et al. [19]	2Tpsm	3Tpr + 1Tpsm	super	super
He-Chen-Zhang's [26]	1Tsm	3Tsm + 3Tpa	super	normal
Gong and Li's [21]	1Tsm	4Tsm + 3Tpa	normal	super
Wang-Chen-Long-Mao's[22]	1Tsm	3Tsm + 3Tpa	normal	super
Yeh-Su-Choo's [23]	1Tsm	3Tsm + 2Tpa	insecure	normal
Jia-He-Liu-Kim's [24]	1Tsm	4Tsm +2 Tpa	insecure	super
Karati-Islam-Biswas [25]	1Tsm	3Tsm + 3Tpa	super	normal
Du-Wen-Zhang-Gao [13]	1Tsm	4Tsm + 2Tpa	super	super

When evaluating table 2, as security is the most important aspect, it would show that the scheme Du-Wen-Zhang-Gao [13] is a top candidates for deployment in IoT devices. Something to note is that some schemes computationally perform better e.g. Karati-Islam-Biswas [25] the security of this scheme is weaker making these a less suitable candidate for employment.

3.4 Comparison RSA vs DSA vs ECC

Looking at key sizes, which are also often related to the signature size, for signature schemes is important for deployment in IoT devices as these often give an indication on the amount off storage space needed. Storage space and computation cost in time are often related, but for constrained IoT devices it is relevant to also look at storage space as a constrained parameter. The NIST [28] has provided a table where it evaluates the estimated maximum security level of symmetric key algorithms in bits and what the equivalent key sizes for RSA and ECC schemes are. The security level b can thus be seen as 2^b operations needed to break the scheme. Note for DSA p is the prime and q is a prime divisor of p-1.

Table 3. Security level and their RSA, DSA and ECC key sizes in bits [28]

Security level	RSA key size	DSA key size	ECC key size
80	1024	p=1024, q=160	160-223
112	2048	p=2048, q=224	224-255
128	3072	p=3072, q=256	256-383
192	7680	p=7680, q=384	384-511
256	15360	p=15360, q=512	>512

For ECC the equivalent security level can only be reached if the key size is twice as large in bits. However with the way that elliptical curves are generated the lower bound on possible

the equivalent can not necessarily be reached, thus leading to a range for the size.
In [29] it is shown that RSA is outperformed by ECC in terms of total computation cost for security levels of 112 and above. For a 64 bit message ECC outperform RSA by 17.5% at the security level of 112 and 52% at level 128. And for a 256 message ECC outperform RSA by 35.3% at the security level of 112 and 59.2% at level 128. In [11] it is shown that SCDSA can compete with ECC as it is shown to outperform ECDSA in verification time and signature size. Evaluating the signature sizes, RSA (1024 bit key) comes to 1048 bit, ECDSA comes to 832 bits, the Du-Wen-Zhang-Gao CLS [13] scheme comes to 328 bits and SCDSA comes to 320 bit signatures. Note that Du-Wen-Zhang-Gao CLS and SCDSA thus come to a very similar signature size. Even though from table 3. it can be seen that ECC can work with smaller key sizes than compared to RSA and DSA. With ECC using less than 30% the bits RSA or DSA needs for the same security level.

3.5 Suitability for IoT devices

As seen in the comparison SCDSA and the Du-Wen-Zhang-Gao CLS schemes outperform RSA in terms of computation costs and signature size. Although RSA is still viable for more robust IoT devices, RSA is less recommended for smaller IoT devices due to the existence of other more suitable options.
The choice for SCDSA for IoT devices would be suitable. As SCDSA was actually proposed as a lightweight scheme also for use in IoT it has some properties suitable for smaller IoT devices. Althought SCDSA does use a large prime number n, it's low computation cost and small signature size allows it to still be suitable for IoT.
ECC is due to it's inherently small key size and signature size in term of space very suitable for small IoT devices. And owing to the optimizations and thus efficiency increase the Du-Wen-Zhang-Gao CLS scheme is quite suitable for IoT devices.

4 Post-Quantum IoT security

With the rise and development of quantum computers many widely used public key cryptographic schemes could become vulnerable to attacks. For example An adversary with a large universal quantum computer which has Shor's algorithm [30] implemented could break the integer factorization problem in polynomial time on which many schemes such as the RSA rely for their security. HBS schemes base their security on hash functions which have well known security notions. For HBS schemes the most prominent way of attacking the schemes comes from the Grover algorithm [31]. The Grover algorithm is a quantum algorithm used to find a collision that satisfies certain conditions. This algorithm can thus be used to find the original value also called preimage from a hash function. In this section however some schemes are investigated which offer a sufficient level of security against this algorithm.

4.1 Lamport Signature scheme

An implementation of Lamport Signature scheme [32] has been proposed for IoT devices [33]. The Lamport Signature scheme claims it's strength in the strength of it's one way hash function instead of algorithmic complexity which bases itself on computational and storage requirements. As the Lamport Signature Scheme does not rely on computational and storage requirements to break the scheme the Lamport scheme is quantum computing

resistant to break.
A short summary of the proposed Lamport Signature: a private key consists of 256 pairs of 256-bit numbers (2 lists of 256 256-bit numbers) generated by a random number generator (RNG). Each number will be hashed by a hash function F() and this creates the public key, this results in both private and public key using 16KiB. For any message a hash-sum is created and with this a signature is created based on picking bits of one of the lists from the key. This signature then consists of a list of 256 256-bit numbers (8KiB). The verification uses the same method but using the public key in creating a 'check list', The signature is hashed and compared to the 'check list'. If the comparison shows equality the verification is confirmed. If not the message has been tampered with or is not from the original sender, leading to the message being deleted.
Something to note about the Lamport Signature scheme is it's characteristic of being one-time signatures scheme (OTS). Which means requiring the updating of the private key by the manufacturer and public key inside the IoT device as each unique key pair should only be used once. Should a key pair be used more than once degradation of the security take place as this opens the scheme up to two-message attacks. Two-message attacks entail finding secret values through the union between the two signatures. Finding these secret values enables the forgery of a signature and thus the breaking the scheme.
The Lamport Signature scheme for IoT devices paper provided some metrics of estimates on storage and time consumption for some popular development boards used in IoT devices. The following computation cost has been calculated for an arduino with a clock speed of 16 MHz [33]. Something to note is that the maximum security level (under assumption of a perfect hash function) against the Grover algorithm is half of the segments lengths. So for Lamport with the 256 bit numbers this comes to a maximum security level of 128 bits.

Table 4. Metrics on the Lamport scheme.

Scheme	Keysize	Computation cost	Security level
Lamport	16 KiB	0.359s	128

4.2 Winternitz one time signature scheme

The Winternitz one time signature (W-OTS) scheme was proposed by Robert Winternitz in 1979 as mentioned in [34], however first fully described in [35], as an improvement over the Lamport scheme in terms of space which however does trade in time complexity for this (W-OTS is able to reduce the key size by a factor of 4 to 8). As also seen in the Lamport scheme a hash function F() is used. A private key x_i for i = 1, ..., N where N = $\lceil n/t \rceil + \lceil (\lceil log_2 n log_2 t \rceil + t/t \rceil$. The Winternitz scheme does however with the use of private key x and the Winternitz parameter w, which in [35] satisfies $w = 2^t$, calculate the public key Y as $y_i = F^{2^{w-1}}(x_i)$ with Y = F($y_1||y_2||...||y_N$), instead of $y = F(x)$ as in the Lamport scheme. Note that $F^w(x)$ indicates the iterative application of F() on x as ..F(F(F(X))). For the signing the message m of length n is hashed and split into $\lceil n/t \rceil$ segments (with '0' padding if necessary) with length t bits each, resulting in $m_1, m_2, ..., m_{\lceil n/t \rceil}$. Each of these segments are then treated as integers and a check symbol/ checksum is formed. C = $\sum_{i=1}^{\lceil n/t \rceil} 2^t - m_i$ which is important for security reductions. The signature is then generated as $s_i = F^{m_i}(x_i)$ for i = 1, ..., N.
For verification the same method is used as in the signing to get $m_1, m_2, ..., m_{\lceil n/t \rceil}$. Then a V is calculated using $v_i = F^{2^t - m_i - 1}(s_i)$ resulting in $V = F(v_1, v_2, ..., v_N)$ and lastly the signature is verified if V = Y.

4.2.1 W-OTS+

Some variants exist for the Winternitz Scheme. In this section two variants are discussed. The scheme for convenience called W-OTS^{prf} [36] (prf for pseudo random functions) which is used in the eXtended Merkle Signature Scheme discussed in 4.3. And the W-OTS+ scheme [37] which was proposed as a scheme that has even smaller signature sizes than the Winternitz Scheme and it's variants. Both schemes have a function family F_n with functions denoted as f_k, where k is the function key, both also use a Winternitz parameter w $>$ 1 and message length m to compute an l with $l_1 = \lceil m/log_2(w) \rceil$, $l_2 = \lfloor log_2(l_1(w-1))/log_2(w) \rfloor + 1$, giving $l = l_1 + l_2$.

W-OTS^{prf} contrasts with the original W-OTS scheme in that it does not use thee functions in an iterative manner. Instead the result of one $f_k(r)$ is used as key for the next application of a function and the input r stays the same, meaning $f_k^2(x) = f_{f_k(x)}(x)$. Furthermore the schemes relies on the function family being a family of pseudo random functions, hence the name W-OTS^{prf}. The private key $x = (x_1, ..., x_i)$ for i = 1, ..., l is chosen random with each segment/bit string being of length n. The public key Y is then calculated with random number r as $y_i = f_{x_i}^{w-1}(r)$ for i = 1, ..., l and $Y = (r, y_1, ..., y_l)$. The message is also divided into l segments $M = (m_1, ..., m_{l_1})$ and for signing a checksum is computed C = $\sum_{i=1}^{l_1}(w - 1 - m_i)$. This checksum can be represented as $C = (C_1, ..., C_{l_2})$. With the checksum and the message a B is set which is $B = (b_1, ..., b_l) = M||C$. Lastly the signature is made $S = (s_1, ..., s_l) = (f_{x_1}^{b_1}(r), ..., f_{x_l}^{b_l}(r))$. For verification the same checksum and B are computed and $V = (f_{s_1}^{w-b_1-1}(y_1), ..., f_{s_l}^{w-b_l-1}(y_l))$ is computed. The signature is verified if $V = (y_1, ..., y_l)$.

For W-OTS+ a chaining function is defined, this function uses a random $r = (r_1, ..., r_j)$ (Note $j \geq i$), in the function $c_k^i(x, r) = f_k(c_k^{i-1}(x, r) \oplus r_i)$ for i $>$ 0 (if i = 0 then $c_k^0(x, r) = x$). In the key generation algorithm $l + w - 1$ n-bit strings are randomly generated. The first l n-bit strings forms the private key $x = (x_1, ..., x_l)$ and the last w - 1 n-bit strings form the 'randomization elements' $r = (r_1, ..., r_{w1})$. Note $r_{a,b}$ denotes the subset from r from element r_a to r_b where if a $>$ b $r_{a,b}$ is an empty string. Furthermore a random function key k is chosen. With this the public key is computed $Y = ((r, k), c_k^{w-1}(x_1, r), ..., c_k^{w-1}(x_l, r))$. The signing and verification are pretty similair as in W-OTS^{prf} except for how the functions actually behave. For signing the message is,, divided into w-1 segments $M = (m_1, ..., m_{l_1})$, the checksum is computed C = $\sum_{i=1}^{l_1}(w-1-m_i)$ and $B = (b_1, ..., b_l) = M||C$. The signature is computed $S = (s_1, ..., s_l) = (c_k^{b_1}(x_1, r), ..., c_k^{b_l}(x_l, r))$. And for verification the same B is calculated and $V = ((r, k), c_k^{w-b_1-1}(s_1, r_{b_1+1,w-1}), ..., c_k^{w-b_l-1}(s_l, r_{b_l+1,w-1})$. The signature is verified if V = Y.

4.2.2 W-OTS security level

Both W-OTS+ and W-OTS^{prf} have shown that, if their function family is second key resistant or key collision resistant, to both be existential unforgeable under adaptive chosen message attacks (EU-CMA) and both are even strongly unforgable under adaptive chosen message attacks (SU-CMA). Both schemes also provided formulas for their security level, W-OTS^{prf} security level $b \geq n - w - 1 - 2log_2(lw)$ [36] and W-OTS+ security level $b \geq n - log_2(w^2 * l + w)$. [37] however in [38] the security level was updated to $b \geq n - log(2w^2 l + wl)$, this will be denoted under W-OTS+^{up}. The updated security level difference is almost insignificant to the original the updated version should be known for a fair evaluation. In the W-OTS+ paper a comparison for W-OTS+, W-OTS and W-OTS^{prf} on the parameters for a security level of $b \geq 100$ and a signature size under 1 kB. This comparison used a message length m = 256. Only the signing cost in bytes is included as key generation and verification have the same costs. Furthermore note that W-OTs^{prf} could not satisfy the signature size and security level conditions. l = 62

Table 5. W-OTS schemes comparison [37].

Scheme	n	w	signature size	Signing cost	Security level
W-OTS+	128	21	992	1,302s	113*
W-OTS	256	455	992	14,105	128
W-OTS^{prf}	128	8	1440	0,720s	100

*The updated security level from W-OTS+^{up} also gives 113.

For W-OTS as although the result shown is theoretically possible, getting the signature size under 1 kB is impractical. This is due to the drastic increase of the computation cost because of the need for more function evaluations, which has even bigger impact seeing that n also increased. As can be seen in figure 5 the W-OTS+ scheme provides an adequate (≥ 112) security level for the signature size and signing cost. The W-OTS^{prf} scheme could not get a security level of 100 with an signature size under 1 kB the scheme does have a lower signing cost. So as seen classically in computer science there is a trade-off between speed and space between the two schemes.

4.3 eXtended Merkle Signature Scheme

The Merkle signature scheme [34] is based on hash trees and OTS.

The eXtended Merkle Signature Scheme (XMSS) [39] is based on the Merkle signature scheme and also uses OTS. XMSS enables to produce multiple OTS for a XMSS single keypair. Something to note is that W-OTS^{prf} is used for the OTS in XMSS.

XMSS uses some public parameters: "security parameter" n, a w the Winternitz parameter, the message length m, a function family Fn, height of the tree h, a hash function H() and a random x. XMSS uses a binary tree (for convenience called key-tree) with h + 1 levels where the leaves are on level 0 and the root is at level h. Nodes on level j are denoted by Node$_{i,j}$, 0 â€ i < 2 Hâj. The nodes are constructed using a random bitmask $b = (b_{l,j}||b_{r,j})$ as Node$_{i,j}$ = $H((Node_{2i,j-1} \oplus b_{l,j})||(Node_{2i+1,j-1} \oplus b_{r,j}))$. Then each leaves is constructed with a W-OTS^{prf} public key, thus having 2^h W-OTS^{prf} keys. A leaf is constructed using another tree an L-tree. This L-tree is formed with the property that the leaves are the segments of the public key $x = (x_1, ..., x_l)$. So you only have l leaves, but as l is not necessarily a power of 2 any node that has no right sibling is moved up until it becomes a right sibling. The nodes for the L-tree are constructed the same way as the key-tree but with a new bit mask, this bitmasks is however the same for all l-trees in a key-tree. The key-tree public tree then contains the bitmasks and the root of the tree.

For signing then as there are 2^h W-OTS^{prf} keys available per tree so many messages can be signed using the tree. So for the ith message the ith W-OTS^{prf} key is used. The signature $S = (i, s, AUTH)$. With s the W-OTS^{prf} signature and AUTH the authentication path for the node. This path is the sequence of all the siblings of the path from Node$_{0,i}$ to the root.

For verification the same method as in W-OTS^{prf} for V (without the first element (r,k)) is used. The corresponding leaf/node Node$_{0,i}$ is then constructed using a L-tree. With this leaf and the AUTH the path to the root is computed. If the path computed correctly leads

to the root the signature is accepted.
XMSS allows with this that multiple W-OTS^{prf} verifications can be done on one public XMSS key. Something to note is it that the use of W-OTS^{prf} in XMSS could also be replaced with the use of W-OTS+. This change can lead to the improvement of the security level of XMSS as shown in [?].

4.4 Quantum resistant schemes in IoT

The presented quantum resistant schemes are in general suitable for use IoT. The Lamport scheme with the largest key size of 16 kiB is not suitable for the smallest IoT devices which have ≤ 16 kiB available. The lamport scheme is however viable for IoT devices with more storage capacity available. The W-OTS scheme and their variants are quite suitable for use in IoT as they have smaller key sizes. Although the W-OTS scheme does trade in efficiency somewhat the variants do make up somewhat of that with their optimizations. XMSS is suitable for IoT devices with [40] even showing that speedups through hardware can be reached.

5 Responsible Research

As this paper is a study into existing schemes and thus extracts data and results from these the main ethical aspect to consider is plagiarism. Each piece of data that has been taken from a paper has an appropriate reference to the corresponding paper. Another risk contained in this paper is the coercion of picking a certain scheme over another based on the comparison contained. This paper tries to present results as objective as possible. However some metrics have been generated under different parameter settings, which could lead to a skewed vision between schemes and their efficiencies. Any manufacturer of IoT devices should be aware that actual implementations on devices may vary from the results presented in this paper.

6 Discussion

As some of the metrics found for the computation costs of the different schemes have not been produced in the same environment, these metrics are not directly comparable to each other. Which also gives that some schemes are not directly comparable but need another reference point, which for example is seen for the security levels. Because the principles on which the signature schemes base their security off are different, the use of for example the security levels provided by the NIST are used. With these guidelines it is easier to estimate and compare the security levels between schemes. As although the presented schemes do differ somewhat to their overarching classification by NIST the guidelines are a useful baseline.

7 Conclusions and Future Work

From the presented schemes and the comparison between them it is shown that suitable options for signature schemes in IoT devices, such as SCDSA and the Du-Wen-Zhang-Gao CLS scheme, are currently available. Due to the development of quantum computers the improvement to quantum resistant schemes seems inevitable. This necessary change is supported for IoT as quantum resistant schemes that are suitable for use in IoT devices have been presented.

This paper does not contain a comparison for all security reductions on which signature schemes can be based. For example in the post-quantum group Lattice based cryptography such as NTRU, Multivariate based cryptography such as Rainbow and Supersingular elliptic curve isogeny cryptography such as Supersingular Isogeny Diffie-Hellman (SIDH). Al thought these schemes are not necessarily more efficient or even suitable for use in IoT. It could be relevant to also look at theses schemes and their different security reductions. Another interesting thing to investigate would be implementations for hybrid for pre- and post-quantum schemes. Although in the future, when quantum computing has progressed far enough, the switch to full quantum resistant schemes is inevitable. The transition from the pre-quantum to post-quantum could be greatly helped by having a hybrid scheme as an intermediate step.

A Appendix

A.1 References

References

[1] Statista. (2021). *Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030.* Retrieved 2021-04-23, from https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[2] Posey, B. (2021) *IoT devices (internet of things devices).* Retrieved 2021-04-18, from https://internetofthingsagenda.techtarget.com/definition/IoT-device

[3] Subahi, Anoud Theodorakopoulos, George. (2019). Detecting IoT User Behavior and Sensitive Information in Encrypted IoT-App Traffic. *Sensors* (Basel, Switzerland). 19. 10.3390/s19214777.

[4] Tang, Q., Du, T. (2021). Internet of Things Security: Principles and Practice. *Springer, Singapore.*

[5] NXP. (2020) *Security Primitives: Common Nomenclature to Describe Security Requirements in (I)IoT Systems.* Retrieved 2021-05-04, from https://www.nxp.com/docs/en/white-paper/SEC_PRIMITIVES_WP.pdf

[6] T. M. FernÃ¡ndez-CaramÃ©s, From Pre-Quantum to Post-Quantum IoT Security: A Survey on Quantum-Resistant Cryptosystems for the Internet of Things, in *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6457-6480, July 2020, doi: 10.1109/JIOT.2019.2958788.

[7] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Trans. Inf. Theory,* vol. IT-31, no. 4, pp. 469-472, Jul. 1985.

[8] Rivest, R. L., Shamir, A., Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120-126.

[9] Barker, Elaine; Dang, Quynh (2015-01-22). NIST Special Publication 800-57 Part 3 Revision 1: Recommendation for Key Management: Application-Specific Key Management Guidance (PDF). *National Institute of Standards and Technology: 12.* doi:10.6028/NIST.SP.800-57pt3r1. Retrieved on: 2021-06-01

[10] Zimmerman, P. (2020, Feb 28). *Factorization of RSA-250.* https://lists.gforge.inria.fr/pipermail/cado-nfs-discuss/2020-February/001166.html

[11] M. A. Mughal, X. Luo, A. Ullah, S. Ullah and Z. Mahmood, "A Lightweight Digital Signature Based Security Scheme for Human-Centered Internet of Things," in *IEEE Access*, vol. 6, pp. 31630-31643, 2018, doi: 10.1109/ACCESS.2018.2844406.

[12] S. S. Al-Riyami and K. G. Paterson, (2003, November). Certificateless public key cryptography. In *International conference on the theory and application of cryptology and information security* (pp. 452-473).

[13] Du, H., Wen, Q., Zhang, S., Gao, M., A new provably secure certificateless signature scheme for Internet of Things, *Ad Hoc Networks*, Volume 100, 2020, 102074, ISSN 1570-8705, doi:10.1016/j.adhoc.2020.102074.

[14] Johnson, D., Menezes, A., Vanstone, S. (2001). The elliptic curve digital signature algorithm (ECDSA). *International journal of information security*, 1(1), 36-63.

[15] Y.-C. Chen and R. Tso, A survey on security of certificateless signature schemes, *IETE Tech.* Review, vol. 33, no. 2, pp. 115-121, 2016. doi: 10.1080/02564602.2015.1049223

[16] Huang, X., Mu, Y., Susilo, W., Wong, D. S., Wu, W. (2007, July). Certificateless signature revisited. In Australasian Conference on Information Security and Privacy (pp. 308-322). *Springer, Berlin, Heidelberg.*

[17] Z. Zhang, D. S. Wong, J. Xu, D. Feng, (2006, June). Certificateless public-key signature: security model and efficient construction. In *International Conference on Applied Cryptography and Network Security* (pp. 293-308). Springer, Berlin, Heidelberg.

[18] Choi, K. Y., Park, J. H., Lee, D. H. (2011). A new provably secure certificateless short signature scheme. *Computers Mathematics with Applications*, 61(7), 1760-1768.

[19] Chen, Y. C., Tso, R., Horng, G., Fan, C. I., Hsu, R. H. (2015). Strongly Secure Certificateless Signature: Cryptanalysis and Improvement of two Schemes. *J. Inf. Sci. Eng.*, 31(1), 297-314.

[20] He, D., Chen, J., Zhang, R. (2012). An efficient and provablyâsecure certificateless signature scheme without bilinear pairings. *International Journal of Communication Systems*, 25(11), 1432-1442.

[21] Gong, P., Li, P. (2014). Further improvement of a certificateless signature scheme without pairing. *International Journal of Communication Systems*, 27(10), 2083-2091.

[22] Wang, L., Chen, K., Long, Y., Mao, X., Wang, H. (2015, September). A modified efficient certificateless signature scheme without bilinear pairings. In *2015 International Conference on Intelligent Networking and Collaborative Systems* (pp. 82-85). IEEE.

[23] Yeh, K. H., Su, C., Choo, K. K. R., Chiu, W. (2017). A novel certificateless signature scheme for smart objects in the Internet-of-Things. *Sensors*, 17(5), 1001.

[24] Jia, X., He, D., Liu, Q., Choo, K. K. R. (2018). An efficient provably-secure certificateless signature scheme for Internet-of-Things deployment. *Ad Hoc Networks*, 71, 78-87.

[25] Karati, A., Islam, S. H., Biswas, G. P. (2018). A pairing-free and provably secure certificateless signature scheme. *Information Sciences*, 450, 378-391.

[26] D. He, S. Zeadally, B. Xu, X. Huang. "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Trans. Inf. Forensics Secur.*, 10 (12) (2015), pp. 2681-2691

[27] MIRACL library (2019) MIRACL [Source code]. https://github.com/miracl/MIRACL/

[28] Barker, E. (2016-01-28). "NIST Special Publication 800-57 Part 1 Revision 4: Recommendation for Key Management: General" (PDF). *National Institute of Standards and Technology*: 53. doi:10.6028/NIST.SP.800-57pt1r4.

[29] Mahto, Dindayal Khan, Danish YADAV, DILIP. (2016). Security Analysis of Elliptic Curve Cryptography and RSA.

[30] P. W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124-134, doi: 10.1109/SFCS.1994.365700.

[31] Grover, L.K.: A fast quantum mechanical algorithm for database search. In: *Proceedings of ACM STOC 1996*. pp. 212â219. ACM, New York, NY, USA DOI:https://doi-org.tudelft.idm.oclc.org/10.1145/237814.237866

[32] Lamport, L.: Constructing digital signatures from a one way function. *Technical Report SRI-CSL-98, SRI International Computer Science Laboratory*, 1979

[33] G. M. Abdullah, Q. Mehmood and C. B. A. Khan, Adoption of Lamport signature scheme to implement digital signatures in IoT, *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2018, pp. 1-4, doi: 10.1109/ICOMET.2018.8346359.

[34] Merkle, Ralph. (1989). A Certified Digital Signature. 435. 218-238. 10.1007/0-387-34805-0_21.

[35] Dods, C., Smart, N. P., Stam, M. (2005, December). Hash based digital signature schemes. *In IMA International Conference on Cryptography and Coding* (pp. 96-115). Springer, Berlin, Heidelberg.

[36] Buchmann, Johannes Dahmen, Erik Ereth, Sarah Hülsing, Andreas Rückert, Markus. (2011). On the Security of the Winternitz One-Time Signature Scheme. *International Journal of Applied Cryptography*. 3. 363-378. 10.1007/978-3-642-21969-6_23.

[37] Hülsing, A. T. (2013). W-OTS+ - shorter signatures for hash-based signature schemes. In A. Youssef, A. Nitaj, A. E. Hassanien (Eds.), *Progress in CryptologyâAFRICACRYPT 2013: 6th International Conference on Cryptology in Africa, Cairo, Egypt*, June 22-24, 2013. Proceedings (pp. 173-188). (Lecture Notes in Computer Science (LNSC); Vol. 7918). Springer. https://doi.org/10.1007/978-3-642-38553-7_10

[38] Kudinov, Mikhail Kiktenko, Evgeniy Fedorov, A.. (2020). Security analysis of the W-OTS$^+$ signature scheme: Updating security bounds.

[39] Buchmann, J., Dahmen, E., HÃŒlsing, A. (2011, November). XMSS-a practical forward secure signature scheme based on minimal security assumptions. *In International Workshop on Post-Quantum Cryptography* (pp. 117-129). Springer, Berlin, Heidelberg.

[40] Ghosh, S., Misoczki, R., Sastry, M. R. (2019). Lightweight Post-Quantum-Secure Digital Signature Approach for IoT Motes. IACR Cryptol. ePrint Arch., 2019, 122.