

Fast Verification of Digital Signatures in IoT

Apurva S. Kittur^(✉), Ashu Jain, and Alwyn Roshan Pais

Information Security and Research Lab, Department of Computer Science
and Engineering, National Institute of Technology Karnataka,
Surathkal, Karnataka, India
apurva.kittur@gmail.com, ashurr99@gmail.com, alwyn.pais@gmail.com

Abstract. Internet of Things (IoT) is the recent advancement in Wireless technology where multiple embedded devices are connected through internet for exchange of information. Since the information exchanged is private and at times confidential, state of the art focusses at providing proper security to the system. To avoid illegal users from getting access to information system, authentication through Digital Signatures becomes integral part of IoT. Verifying individual signatures is a time consuming process, hence it is not advisable in IoT systems. Using Batch verification of Digital signatures, reduction in verification time is achievable. Hence in this paper, we have studied different RSA based batch verification techniques and their analysis is provided. Batch verification of digital signatures in IoT devices is a promising area for further research.

1 Introduction

Internet of Things (IoT) was coined in 1999 by Kevin Ashton. ‘Internet’ refers to the interconnectivity of devices to create a network, and ‘Things’ refers to the objects or devices that have the capability to connect to the Internet. The Internet of Things (IoT) can be defined in many ways [2, 10, 15, 31]. One way of defining can be, ‘it is a network of sensors and smart devices which sense the data which is further processed and analysed in a ubiquitous network.’ IoT has seen rapid development in recent years because of its ‘smartness’. The various applications of IoT include Smart City [5, 17], Smart Home [6, 16], and Smart Health [1] etc. These applications have millions of devices generating large volumes of data.

As we know the sensors are used for monitoring various physical conditions like temperature, sound, pressure etc. The network of these several distributed sensing objects are collectively referred as Wireless Sensor Network (WSN). These WSN nodes are deployed largely in various applications because of their low cost and low power consumption. WSN edge nodes act as gateways or bridge between sensors and internet protocol as depicted in Fig. 1. These gateway nodes collect data from the sensor nodes, and normalize the information received for further processing and storage and they are also responsible for providing security. These nodes initially authenticate the sensor node before the exchange of data. These set of edge nodes together have more energy and computation power

for processing than individual sensor nodes. Hence they play the role of firewall by providing the security to sensor nodes as well as to the internet protocol.

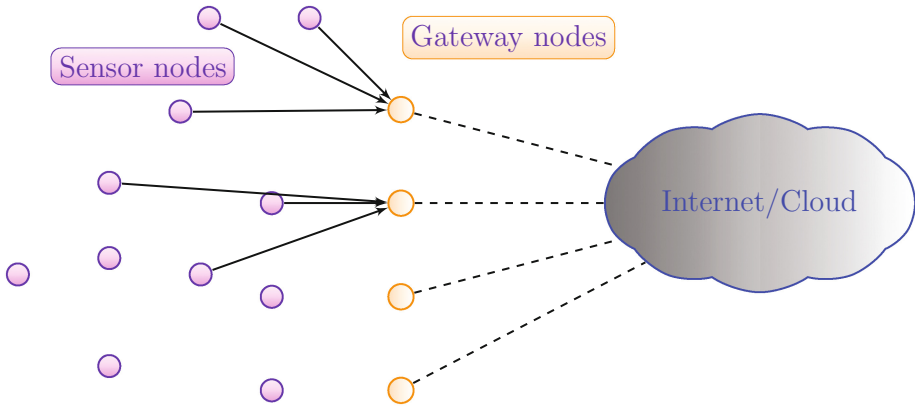


Fig. 1. Basic structure of IoT

1.1 Security in IoT

Security is the major concern in IoT since millions of devices sense and communicate large volumes of private and sensitive data. There are a number of fundamental security capabilities that a IoT system should possess, since the sensor nodes are more vulnerable to threats. Therefore IoT security standard must address the challenges of scalability, privacy and authentication etc. IoT is a combination of various networks, where various sensor nodes generate heterogeneous sets of data. Therefore building a standard secure and reliable system for IoT is still a challenge.

Most of the threats are categorised into three major categories:

- **Capture:** The attacker captures or gets access to the system or information. In the threats like eavesdropping, the attacker tries to obtain control over the system and the private data.
- **Disrupt:** This attack refers to destroying, denying or disturbing the system from proper functioning. Replay attack is one of the examples under this threat.
- **Manipulate:** This attack refers at manipulating critical data, identity etc. Man-in-the-middle attack is an example for the same.

There are various ways to overcome these threats by implementing security protocols such as TLS, SSL, and by providing digital certificate standard and Certificate Authorities (CA), which are based on Public Key Infrastructure (PKI). Before processing any data, the authenticity of the sender has to be verified by

verifying the Digital Signature of the sender. There are many standard Digital signature algorithms introduced such as RSA Digital Signature, DSA, and ECDSA etc. which satisfy the CIA (Confidentiality, Integrity, Authentication) triad properties.

1.2 Batch Verification in IoT

Authenticating every data being exchanged in IoT is a challenge. Individual verification of Digital signatures reduces the performance of the real time IoT system. If the signatures are verified in batches then the verification time can be significantly reduced. Batch verification has two main advantages: one is decreased computation load and the other is reduced computation time at verification side. Hence our study focusses on efficient deployment of Batch verification techniques in IoT system. We also provide results for performance gain over existing system.

As per our understanding, there has been no study on implementing batch verification in IoT. Since IoT nodes have low computation power and memory, batch verification leads to significant increase in performance.

The organisation of the paper is as follows: Sect. 2 throws light on the related research carried out on the topic. In Sect. 3, Harn proposed the standard definitions and in Sects. 4 and 5, we discuss our proposed idea and the results supporting our claim respectively. Section 6 discusses the security analysis of the proposed scheme and we conclude the paper with Sect. 6 and also provide the future scope, followed by references.

2 Related Work

There has been lot of research on the security of IoT in recent times [25, 28–30]. Many researchers have been in to standardizing the security protocols for IoT, but due to its diversity in varied applications, it is difficult to standardize the security architecture. Various lightweight authentication schemes are provided to reduce computation load and computation time [13, 14, 20, 21] on the IoT devices.

There are many Digital Signatures schemes [7, 19, 22, 26] proposed for checking the Authenticity, Integrity and Non-repudiation properties. There has been research on improving the signature verification time through Batch verification [8]. And many Batch verification techniques for RSA Digital signatures [3, 12], DSA signatures [11, 24], ECDSA signatures [27] etc. are proposed. As per our knowledge there is no standard, efficient batch signature verification technique introduced for IoT as of now.

3 Definitions

In this section we provide formal definitions of various notions.

Definition 1. A *Digital Signature Scheme* is actually a systematic study of three probabilistic algorithms (*Gen*, *Sign*, *Vrfy*) [18]:

- *Gen* is the Key Generation algorithm, which takes security parameter 1^n as input and generates the (pk, sk) as output, where pk is public key and sk is private key. We assume that pk and sk each have length at least n , and that n can be determined from pk and sk .
- *Sign* is the Signing algorithm that takes the private key sk and the message m as inputs and outputs signature s , which can be written as $s \leftarrow \text{Sign}_{sk}(m)$.
- *Vrfy* is the Verification algorithm, which takes the public key pk , message m and the signature s as inputs and outputs b whose value is either '1', if the signature is valid and '0', if the signature is invalid. It can be shown as $b \leftarrow \text{Vrfy}_{pk}(m, s)$.

It is required that except with negligible probability over (pk, sk) output by $\text{Gen}(1^n)$, it holds that $\text{Vrfy}_{pk}(m, \text{Sign}_{sk}(m)) = 1$ for every (legal) message m . Signature s is considered valid if $\text{Vrfy}_{pk}(m, s) = 1$

Definition 2. Batch Verification Algorithm: Suppose $(\text{Gen}, \text{Sign}, \text{Vrfy})$ is a Digital Signature Scheme with l as the security parameter, $k, n \in \text{poly}(l)$, $PK = pk_1, \dots, pk_k$ and $(pk_1, sk_1), \dots, (pk_k, sk_k)$ are generated by $\text{Gen}(1^l)$, the Batch Verification Algorithm [4] should hold the following conditions:

- If $pk_i \in PK$ and $\text{Vrfy}_{pk_i}(m_i, s_i) = 1$ for $i \in [1, n]$ then $\text{Batch}((pk_1, m_1, s_1), \dots, (pk_n, m_n, s_n)) = 1$
- If $pk_i \in PK$ for all $i \in [1, n]$ and $\text{Vrfy}_{pk_i}(m_i, s_i) = 0$ for some $i \in [1, n]$, then $\text{Batch}((pk_1, m_1, s_1), \dots, (pk_n, m_n, s_n)) = 0$ except with negligible probability in l , over the randomness of Batch.

4 Proposed Method

As IoT devices have huge information exchange, providing end-to-end authentication between the sensor nodes is very critical. In our work, we have reduced the verification time required for authentication of these millions of nodes in IoT. As we know, batch verification of signatures reduces the total verification time, but in order to further reduce the verification time, we have applied parallelism along with batch verification. As explained earlier, the edge nodes in IoT can distribute the verification and processing load among themselves as in the cluster considered for our study.

Parallel processing has the advantage of reduced computation time and cost. Therefore in our study, we are implementing parallel processing for three batch Verification Algorithms, A1 [12], A2 [23] and A3 [3] signed with RSA digital signature scheme. We use MPI (Message Passing Interface) [9] in order to distribute the load among the different processors in the workstation cluster. MPI provides the specifications for the library for efficient message passing in parallel. MPI specifications provide advantages such as portability, efficiency and flexibility across various platforms.

4.1 Algorithms Considered for Study

For our experimentation, we consider multiple signatures signed by RSA digital signature scheme. There are many techniques proposed for verification of RSA signatures in batches. We have considered three algorithms which were proposed initially which verify the given batch of RSA signatures for the presence of invalid signature. If there is occurrence of invalid signature, then all the signatures in the batch are verified individually to identify the location of that signature. The three algorithms considered for our study are:

Algorithm A1: L. Harn proposed the first scheme for batch verification of RSA Digital Signatures. The message to be sent is first hashed, then signed and the signature generated is appended with the message and sent to the verifier. The equation proposed for signature verification at the verifier is,

$$\left(\prod_{i=1}^t s_i\right)^e = \prod_{i=1}^t h(m_i) \bmod n \quad (1)$$

From the above equation it is clear that, after the receiving the signatures s_i , at the LHS side, all the s_i values are multiplied, and are exponentiated with the public key e . Then on the RHS side, hash values $h(m_i)$ for each message are generated and re multiplied if both the values of LHS and RHS match, all the signatures are valid or else there are one/more invalid signatures existing in the given batch.

Algorithm A2: This algorithm proposed by Hwang et al. is the modification to Algorithm A1, and improves the security over algorithm A1. The proposed equation to batch verify the signatures is,

$$\left(\prod_{i=1}^t s_i^{v_i}\right)^e = \prod_{i=1}^t h(m_i)^{v_i} \bmod n \quad (2)$$

where v_i is a small random number generated at the verifier, which is used as an exponent for verification. And all these signatures are then multiplied and verified. Similar to the first algorithm, if both the values of LHS and RHS match, all the signatures are valid or else there are one/more invalid signatures existing in the given batch.

Algorithm A3: This algorithm is proposed by Bao [3] which makes sure that the signature can be generated only with the valid private key. The verifier makes this slight modification to the Hwang's scheme [23],

$$\left(\prod_{i=1}^t s_i^{v_i}\right)^{2e} = \prod_{i=1}^t h(m_i)^{2v_i} \bmod n, \quad (3)$$

where v_i are random numbers generated by the verifier.

As we know there are three main phases in Digital Signature Algorithms: Key Generation, Signature Generation and Signature Verification. In our scheme, we

are introducing parallelism in Signature verification phase. The signatures are generated for various messages either signed by single device or multiple signers. The batch verification algorithm can be used to verify the signatures signed using the following three Types:

- **Type 1:** Single signer uses his private key (sk) to generate signatures for multiple messages (m_1, m_2, \dots, m_t). The signatures are verified in a batch of t signatures (s_1, s_2, \dots, s_t) at once.
- **Type 2:** Multiple signers use their private keys to sign multiple messages (m_1, m_2, \dots, m_t). Signatures (s_1, s_2, \dots, s_t) are verified using the batch verification algorithm where in the signatures correspond to n different signers ($2 \leq n \leq t$).
- **Type 3:** The signatures which can not be categorized in Type 1 and 2 can be categorized in this Type.

4.2 Hardware Specifications

Our study focuses on Type 1 signatures, since we are considering RSA batch verification techniques efficient for Type 1 signatures. Our analysis yields around 80–85% efficiency with inclusion of 7 workstations working in parallel.

The system considered for experimentation is a Rock cluster 6.0 system. The system has 7 workstations. Each workstation has 2 sockets, and each socket has 10 cores. And each core runs with 2.3 GHz processor. Among these seven workstations, one acts as the master which distributes the load among remaining six slaves using MPI library standard. The computation results of all the workstations running in parallel are aggregated and the final results are displayed by the master. This results in significant reduction in verification time of multiple signatures.

4.3 Workflow

Our aim of the work is to reduce the computation load on single node during signature verification, since IoT sensor nodes have limited capacity. The verification load is distributed among the available nodes through parallel processing which reduces the computation time and load.

In the proposed system for batch verification, server node will perform the task of scheduling the batch verification jobs amongst the available gateway nodes and will generate the final results. To emulate this scenario, we have designed and implemented a 7 node cluster system for the batch verification of digital signatures. It may be noted that each cluster node has large capacity and computation power in comparison to a gateway node. Gateway nodes have either dual or quad core 500 MHz–1 GHz processors. Therefore each processor of our cluster system is equivalent to two Gateway nodes.

In Fig. 2, we can observe that the Master distributes load to other workstations, and the communication happens through MPI. Each workstation gets a set of signatures which have to be verified through batch verification. The public

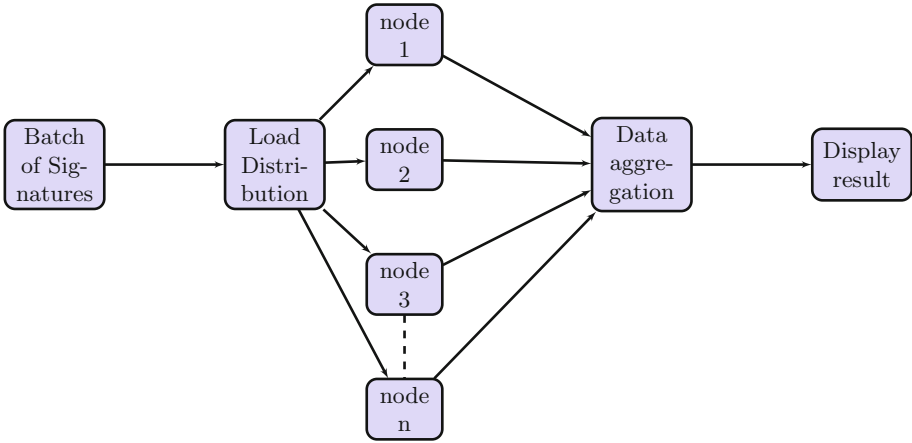


Fig. 2. Workflow of processing

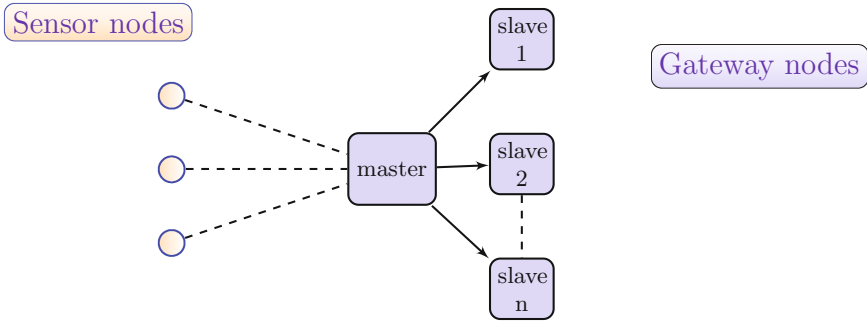


Fig. 3. Signature verification in IoT

key information is shared by the all the workstations. If there is occurrence of an invalid signature/s, the batch verification algorithm at the respective workstation fails. This provides an advantage over serial processing where occurrence of invalid signature involves individual verification of entire batch to identify the faulty signature. In case of parallel verification, the batch size is reduced, therefore number of individual verifications to identify faulty signature/s is reduced.

Figure 3 depicts the scenario of load distribution in IoT. When the gateway node receives batch of digital signatures from sensors/IoT devices, it first identifies other gateway nodes which are available: the ones which have enough power for computation, and the ones which are not very busy in other computations. After figuring the available nodes, it distributes batch of signatures to them. Therefore these available nodes act as slave and the distributing node acts as the master node.

Gateway nodes have more computing power than the sensors or IoT devices, every Gateway node can almost process data from around 2000 sensors. Therefore to handle more load i.e., to process more data from sensors, higher processing power is needed.

5 Results

We have implemented three Batch verification algorithms and analysed their results. We provide the results for batch sizes of $2^4, 2^8, 2^{12}, 2^{16}, 2^{20}$, running in parallel on a cluster consisting of seven nodes. Each node consists of two CPUs with 20 cores. Therefore our system with seven nodes is cluster of 140 cores. We also provide the verification time when the same batch of signatures are verified without parallel processing with MPI.

Case 1: For algorithm A1, the details of time required are given in the Table 1.

Table 1. Verification Time(sec) for Algorithm A1

Batch size	Individual verification	No. of cluster nodes						
		1	2	3	4	5	6	7
2^4	0.03	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029	0.0029
2^8	0.28	0.174	0.1296	0.113	0.09	0.079	0.069	0.069
2^{12}	3.83	0.3158	0.1698	0.1109	0.0854	0.0632	0.0605	0.0565
2^{16}	60.21	3.682	2.061	1.3981	1.1204	0.8744	0.7214	0.6354
2^{20}	970.22	61.6181	31.1445	20.7126	16.2244	13.2463	10.8428	7.0895

The verification time obtained for Algorithm A1 are shown in Table 1. The Table clearly indicates, as the number of workstations increases, the verification time required for the batch of signatures subsequently reduces. It can also be seen that as the batch size of signatures increases, the verification time also increases accordingly. We can also observe the perform gain. The verification time for batch size 2^4 remains almost same for all seven machines is because the amount of time needed for verification of such small batch size very less.

Case 2: For Algorithm A2, the details of time required are given in the Table 2.

Table 2. Verification Time(sec) for Algorithm A2

Batch Size	Individual Verification	No. of Cluster nodes						
		1	2	3	4	5	6	7
2^4	0.003	0.0057	0.0057	0.0057	0.0057	0.0057	0.0057	0.0057
2^8	0.03	0.2369	0.136	0.11	0.101	0.075	0.07	0.07
2^{12}	4.07	0.336	0.1773	0.1162	0.0852	0.0717	0.0709	0.0605
2^{16}	64.60	3.9138	2.2428	1.5071	1.2114	0.9363	0.8036	0.7213
2^{20}	1029.17	62.4667	33.0693	21.5816	17.0072	13.6976	11.5284	9.3254

Table 2 shows the results obtained for Algorithm A2, for the same input given. For 7 machines, the performance gained is almost 6 - 6.5 times. There is very little difference in the increased time for verification for this algorithm since the number of modular exponentiations increases, but the difference is negligible when compared to the security provided.

Case 3: For algorithm A3, the details of time required are given in the Table 3.

Table 3. Verification Time(sec) for Algorithm A3

Batch Size	Individual Verification	No. of Cluster nodes						
		1	2	3	4	5	6	7
2^4	0.03	0.0075	0.0075	0.0075	0.0075	0.0075	0.0075	0.0075
2^8	0.27	0.0276	0.0147	0.0114	0.011	0.00914	0.00815	0.00815
2^{12}	4.05	0.3538	0.1803	0.1265	0.1051	0.0722	0.07147	0.06474
2^{16}	64.48	3.9312	2.2514	1.6943	1.3436	0.9506	0.9006	0.8036
2^{20}	1025.07	62.5376	34.7386	23.0379	17.1628	13.956	12.1236	10.6987

Table 3 for Algorithm A3 has similar results to show. There is no much difference in the number of exponentiation operations when compared to Algorithm A2, but Algorithm A3 is more secure.

6 Security Analysis

Since our study focuses on three Batch verification techniques for RSA digital signatures, in this section we analyse the security aspects of the three techniques and compare them. The algorithm A1 by L. Harn is prone to adaptive chosen message attack. This can be explained as follows, If an attacker wants to send a set of messages m_1, m_2, \dots, m_t , he first generates fake signatures for the messages s'_1, s'_2, \dots, s'_t such that $s'_i = s_i * a_i \bmod q$ where $i = 1, 2, \dots, t$ and $\prod_{i=1}^t a_i = 1$ and sends across. Therefore at the verification, these set of signatures get verified successfully and the verifier fails to detect the fake signatures.

In other attack, the sender generates signatures $s'_1 = h(m_3)^d, s'_2 = h(m_1)^d, s'_3 = h(m_2)^d$ etc., which when verified in batch gets successfully verified. But in case of both the attacks, the invalid signatures are identified if verified individually.

To improve the security of algorithm A1, algorithm A2 was introduced. This technique was introduced to overcome the security flaws from the previous technique. But this technique too is vulnerable to attacks. The chances of verifying an invalid signature as valid is 50%. A dishonest signer chooses a w such that $w^2 = 1 \bmod n$ and generates the invalid signatures $s'_i = s_i * w \bmod n$. The probability of choosing an even random number is 50%. Therefore the probability of accepting an invalid signature as valid is 50%.

This technique increases the number of modular exponentiation operations for batch verification at the verifier. Therefore the extra security comes at a

small computation cost. Therefore for a small increase of 2% computation time, we achieve extra security.

Algorithm A3 was introduced to further reduce the possibility of attacks on algorithm A2. This algorithm takes care of the attack shown in previous algorithm, but introduces a constant which slightly increases the computation time compared to the previous algorithm. Since it introduces a constant integer in the exponentiation, there is no significant increase in computation time.

7 Conclusion and Future Scope

As we know that IoT has millions of sensor devices sending information across the network, there is a need to provide security and authentication to prevent the integrity and the privacy of information. Therefore our idea of accelerating the Batch verification techniques, significantly reduces the time needed to verify millions of signatures, which is a significant advantage to the Digital world. This aids for ‘smart’ projects such for smart city, smart healthcare etc.

For our experimental results, we have considered the batch verification techniques introduced for RSA Digital Signature Scheme since it is the first scheme introduced for batch verification strategy and easy to interpret. We extend our experimental results for various batch verification techniques introduced for DSA and ECDSA. And we are looking forward to implement and study batch verification strategy for Type 2 signatures for verification.

References

1. Amendola, S., Lodato, R., Manzari, S., Occhiuzzi, C., Marrocco, G.: RFID technology for IoT-based personal healthcare in smart spaces. *IEEE Internet Things J.* **1**(2), 144–152 (2014)
2. Atzori, L., Iera, A., Morabito, G.: The internet of things: a survey. *Comput. Netw.* **54**(15), 2787–2805 (2010)
3. Bao, F., Lee, C.-C., Hwang, M.-S.: Cryptanalysis and improvement on batch verifying multiple rsa digital signatures. *Appl. Math. Comput.* **172**(2), 1195–1200 (2006)
4. Bellare, M., Garay, J.A., Rabin, T.: Fast batch verification for modular exponentiation and digital signatures. In: Nyberg, K. (ed.) *EUROCRYPT 1998*. LNCS, vol. 1403, pp. 236–250. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0054130>
5. Cocchia, A.: Smart and digital city: a systematic literature review. In: Dameri, R.P., Rosenthal-Sabroux, C. (eds.) *Smart City*. PI, pp. 13–43. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06160-3_2
6. Du, K.-K., Wang, Z.-L., Hong, M.: Human machine interactive system on smart home of IoT. *J. China Univ. Posts Telecommun.* **20**, 96–99 (2013)
7. Even, S., Goldreich, O., Micali, S.: On-line/off-line digital signatures. *J. Cryptol.* **9**(1), 35–67 (1996)
8. Fiat, A.: Batch RSA. In: Brassard, G. (ed.) *CRYPTO 1989*. LNCS, vol. 435, pp. 175–185. Springer, New York (1990). <https://doi.org/10.1007/0-387-34805-0-17>

9. Gropp, W., Lusk, E., Doss, N., Skjellum, A.: A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Comput.* **22**(6), 789–828 (1996)
10. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of things (IoT): a vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **29**(7), 1645–1660 (2013)
11. Harn, L.: Batch verifying multiple DSA-type digital signatures. *Electron. Lett.* **34**(9), 870–871 (1998)
12. Harn, L.: Batch verifying multiple RSA digital signatures. *Electron. Lett.* **34**(12), 1219–1220 (1998)
13. Hernandez-Ramos, J.L., Pawlowski, M.P., Jara, A.J., Skarmeta, A.F., Ladid, L.: Toward a lightweight authentication and authorization framework for smart objects. *IEEE J. Sel. Areas Commun.* **33**(4), 690–702 (2015)
14. Jan, M.A., Nanda, P., He, X., Tan, Z., Liu, R.P.: A robust authentication scheme for observing resources in the internet of things environment. In: 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 205–211. IEEE (2014)
15. Jia, X., Feng, Q., Fan, T., Lei, Q.: Rfid technology and its applications in internet of things (IoT). In: 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), pp. 1282–1285. IEEE (2012)
16. Jie, Y., Pei, J.Y., Jun, L., Yun, G., Wei, X.: Smart home system based on IoT technologies. In: 2013 Fifth International Conference on Computational and Information Sciences (ICCIS), pp. 1789–1791. IEEE (2013)
17. Jin, J., Gubbi, J., Marusic, S., Palaniswami, M.: An information framework for creating a smart city through internet of things. *IEEE Internet Things J.* **1**(2), 112–121 (2014)
18. Katz, J., Lindell, Y.: *Introduction to Modern Cryptography*. CRC Press, Boca Raton (2014)
19. Lamport, L.: Constructing digital signatures from a one-way function. Technical report CSL-98, SRI International Palo Alto (1979)
20. Lee, J.-Y., Lin, W.-C., Huang, Y.-H.: A lightweight authentication protocol for internet of things. In: 2014 International Symposium on Next-Generation Electronics (ISNE), pp. 1–2. IEEE (2014)
21. Liu, J., Xiao, Y., Chen, C.P.: Authentication and access control in the internet of things. In: 2012 32nd International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 588–592. IEEE (2012)
22. Merkle, R.C.: Method of providing digital signatures, US Patent 4,309,569, 5 January 1982
23. Min-Shiang, H., Cheng-Chi, L., Yuan-Liang, T.: Two simple batch verifying multiple digital signatures. In: Qing, S., Okamoto, T., Zhou, J. (eds.) *ICICS 2001*. LNCS, vol. 2229, pp. 233–237. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45600-7_26
24. Naccache, D., M'Raihi, D., Vaudenay, S., Rphaeli, D.: Can D.S.A. be improved? — Complexity trade-offs with the digital signature standard —. In: De Santis, A. (ed.) *EUROCRYPT 1994*. LNCS, vol. 950, pp. 77–85. Springer, Heidelberg (1995). <https://doi.org/10.1007/BFb0053426>
25. Riahi, A., Challal, Y., Natalizio, E., Chtourou, Z., Bouabdallah, A.: A systemic approach for IoT security. In: 2013 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), pp. 351–355. IEEE (2013)
26. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **21**(2), 120–126 (1978)

27. Shao, Z.: Batch verifying multiple DSA-type digital signatures. *Comput. Netw.* **37**(3), 383–389 (2001)
28. Xu, T., Wendt, J.B., Potkonjak, M.: Security of IoT systems: design challenges and opportunities. In: *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*, pp. 417–423. IEEE Press (2014)
29. Zhang, Z.K., Cho, M.C.Y., Wang, C.W., Hsu, C.W., Chen, C.K., Shieh, S.: Iot security: ongoing challenges and research opportunities. In: *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications (SOCA)*, pp. 230–234. IEEE (2014)
30. Zhao, K., Ge, L.: A survey on the internet of things security. In: *2013 9th International Conference on Computational Intelligence and Security (CIS)*, pp. 663–667. IEEE (2013)
31. Zhu, Q., Wang, R., Chen, Q., Liu, Y., Qin, W.: IoT gateway: Bridging wireless sensor networks into internet of things. In: *2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC)*, pp. 347–352. IEEE (2010)

Security in Computing and Communications

5th International Symposium, SSCC 2017, Manipal,

India, September 13-16, 2017, Proceedings

Thampi, S.M.; Martínez Pérez, G.; Westphall, C.B.; Hu, J.;

Fan, C.-I.; Gómez Mármol, F. (Eds.)

2017, XIX, 424 p. 182 illus., Softcover

ISBN: 978-981-10-6897-3