2020 International Conference on Internet of Things and Intelligent Applications (ITIA) | 978-1-7281-9301-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/ITIA50152.2020.9312307

A Direct Anonymous Attestation Scheme Based on Mimic Defense Mechanism

Chen Yu
School of Cyber Science and Engineering
Southeast University
Nanjing, China
792319313@qq.com

Liquan chen
School of Cyber Science and Engineering
Southeast University
Nanjing, China
Lqchen@seu.edu.cn

Tianyu Lu
School of Cyber Science and Engineering
Southeast University
Nanjing, China
effronlu@seu.edu.cn

Abstract—**Machine-to-Machine (M2M) communication is a essential subset of the Internet of Things (IoT). Secure access to communication network systems by M2M devices requires the support of a secure and efficient anonymous authentication protocol. The Direct Anonymous Attestation (DAA) scheme in Trustworthy Computing is a verified security protocol. However, the existing defense system uses a static architecture. The "mimic defense" strategy is characterized by active defense, which is not effective against continuous detection and attack by the attacker. Therefore, in this paper, we propose a Mimic-DAA scheme that incorporates mimic defense to establish an active defense scheme. Multiple heterogeneous and redundant actuators are used to form a DAA verifier and optimization is scheduled so that the behavior of the DAA verifier unpredictable by analysis. The Mimic-DAA proposed in this paper is capable of forming a security mechanism for active defense. The Mimic-DAA scheme effectively safeguard the unpredictability, anonymity, security and system-wide security of M2M communication networks. In comparison with existing DAA schemes, the scheme proposed in this paper improves the safety while maintaining the computational complexity.**

Keywords—M2M, anonymous attestation, mimic defense, active defense

I. Introduction

At present, the Internet of Things (IoT) is developing rapidly, with a large number of IoT devices in use, and the work undertaken by IoT terminal devices is becoming more extensive, diverse and sensitive. At the same time, criminals are also trying to collect this sensitive data and use it for illegal purposes, putting IoT systems to a severe security challenge. M2M communication is an important branch of IoT communication business and is also a universal form of application in IoT [1]. Due to the widespread distribution and application of M2M systems, security issues in M2M devices themselves and in M2M networks can be particularly pronounced, making it essential to establish an effective M2M security system [2]. Considering the computing power of M2M devices and the security requirements of M2M systems, trusted computing is introduced to establish a secure access mechanism in M2M communications [3].

Privacy CA [4] and DAA are anonymous attestation protocols commonly used in trusted computing systems. In the Privacy CA protocol, the anonymity of the platform needs to be guaranteed by a trusted certificate issuing center, and the use of Privacy CA as an attestation protocol in M2M communication systems is inappropriate due to the absence of a mechanism for authenticating remote servers, where fake remote servers can carry out attacks in a form of fake base stations on M2M devices. The mimic defense technology [5] proposed by Wu, a member of the Chinese Academy of Engineering, in 2013 draws on the working mechanism of the biological autoimmune system, which has an endogenous security effect and enhances the unpredictability of the system, thus establishing an endogenous defense system with high reliability, trustworthiness, and availability [6]. This paper combines the mimic defense idea with the DAA protocol to propose a Mimic-DAA scheme, and then an M2M mimic defense system, which attributes the problem of uncertain security threats to M2M network security to problems that can be solved by robust control theory and technology, thus ensuring secure access to M2M devices at the technical architecture level. The application scenario of the anonymous access scheme for M2M networks based on the mimic defense principle is shown in Fig. 1.

Funding Supported: National key research and development program of China (2020YFE0200600)

II. Overview of the DAA scheme

In 2004, Trusted Computing Group (TCG) proposed the DAA [7] to verify the legitimacy of TPM identities. There are four protocol entities in the DAA protocol, which are Issuer, the issuer of the certificate, the Trusted Platform Module (TPM), Host where the TPM resides, and Verifier, the certifier

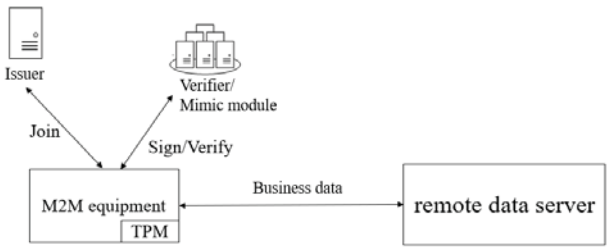

Fig. 1 M2M device communication scenarios

of the signature. The Platform consisting of TPM and Host is also the Signer of the protocol. Five main protocols run between these protocol entities [8], including Setup, Join, Sign, Verify, and Link. The DAA protocol is a three-party mutual attestation process, where any party's illegality can be identified and cause the termination of the protocol's operational process [9]. The Join phase is a mutual attestation process between the platform and Issuer. The Sign and Verify stages are the attestation of the signing party, which blinds the certificate and completes the signature of the message *m* in combination with its own secret value, and the verifying party verifies the legitimacy of the certificate and signature in turn, but does not have access to the identity of the platform in the process [10]. The base name *bsn* is introduced in the Link protocol to provide user-controlled association, and when the same *bsn* signatures are used for a particular platform, it can be inferred that these signatures are from the same platform [11]. The operating mechanism of DAA is shown in Fig. 2.

III. MIMIC-DAA SCHEME

Mimic-DAA optimizes the original DAA scheme by introducing a two-attestation mechanism between the three parties and adding a mimic defense mechanism after the Verify phase. The improved process is shown in Fig. 3.

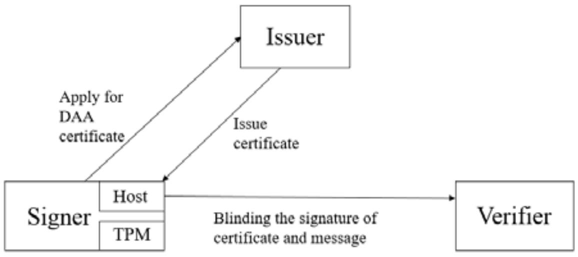
Issuer
Apply for DAA certificate
Issue certificate
Signer
Host
TPM
Blinding the signature of certificate and message
Verifier

Fig. 2 DAA Protocol Operation Mechanism Diagram

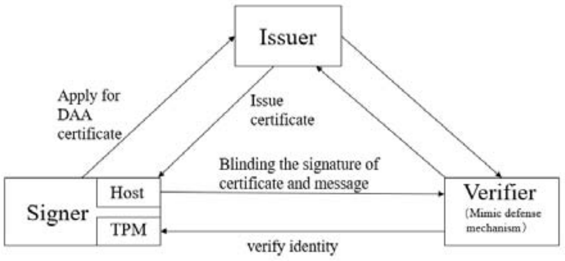
Issuer
Apply for DAA certificate
Issue certificate
Blinding the signature of certificate and message
Signer
Host
TPM
Verifier
(Mimic defense mechanism)
verify identity

Fig. 3 Improved DAA protocol operation mechanism diagram

The symbols used in this paper and their definitions are shown in Table 1.

Table 1 Symbols and definitions in this paper

Symbol	Definition
Z_q	A finite domain of order q
tsk	Private keys held by TPM
tpk	The public key of the TPM, where $tpk = \bar{g}^{tsk}$
$\bar{g}$	Fixed generation elements for all TPMs
bsn_E	The base name of $\hat{g}$, if $bsn_E \neq \perp$, $\hat{g} = H_{G_1}(bsn_E)$, if not $\hat{g} = \bar{g}$
bsn_L	The base name of j , if $bsn_L \neq \perp$, $j = H_{G_1}(bsn_L)$, if not $j = \perp$
m_h	Message Host wants to attach
m_t	Message TPM wants to attach
$\perp$	Special characters for basic names
hsk	The private key held by the Host
π_{cred}	The signature generated by the Host

A. *Initialization settings*

The initialization parameters of the TPM module are generated by calling the TPM command, including the private key $tsk \in Z_q$ and the public key $tpk = \bar{g}^{tsk}$, where $\bar{g}$ is the fixed generating element and the simultaneously internal output parameters are designed. Select the finite loop group of elliptic curves of order three prime numbers G_1 , G_2 , G_T and the private key of Issuer, generate Issuer's public key pair and the hash functions required for subsequent protocols. The initialization algorithm flows as follows.

1) Choose three finite loop groups of elliptic curves of order G_1 , G_2 , G_T with prime q. where $G_1 \neq G_2$ and there is no valid isomorphism from G_2 to G_1 , the generating element of G_1 is g_1 and the generating element of G_2 is g_2 , there is a bilinear mapping $e: G_1 \times G_2 \rightarrow G_T$, and then expose the parameters $(G_1, G_2, G_T, q, g_1, g_2, e)$.

2) Randomly generate the private key $isk: x, y \in Z_q$ for Issuer, calculate the corresponding public key pair $(X, Y): X = g_2{}^x, Y = g_2{}^y$, and then expose the parameter (X, Y) .

3) Call the fixed parameter $\bar{g} = g_1$ of TPM.Create() for each TPM in the domain, expose the public key tpk of the TPM, generate the hash function $H: (0,1)^* \rightarrow (0,1)^l$, $H_{G_1}: (0,1)^* \rightarrow G_1$ needed for each sub-protocol, and expose the hash function (H, H_{G_1}) .

B. *Join protocol*

The Trusted Computing Module is embedded in the M2M device is the TPM entity in the DAA protocol, and the device's own host system corresponds to the Host entity in the DAA protocol, in which the platform can act as both a signatory and a verifier. In TPM, Host and Issuer run the Join protocol, Host acts as an intermediate carrier to complete Issuer's attestation of the platform's identity and receives the DAA certificate sent by Issuer, in which the platform confirms its legal identity by performing zero-knowledge proof of its secret value $gsk = tsk + hsk$, and Host can also verify the legitimacy of the certificate. The process of the protocol is as follows.

1) After receiving the Join message, Issuer randomly generates an integer n of length l to pass to Host.

2) Host selects $\tilde{g} = H_{G_1}(0 \| n)$. Execute the Prove protocol (TPM internal sub-protocol), enter the parameter $(0, tpk, \perp, (0 \| n), ("join", n), \perp)$, and get the output (tpk', π_{tpk}) .

3) Host generates its own private key hsk , calculates $gpk = tpk' \cdot \tilde{g}^{hsk}$, and then sends the $(tpk, tpk', gpk, \pi_{tpk})$ information to Issuer.

4) Issuer first calls $VerSPK(\pi_{tpk}, tpk', \perp, ("join", n), \perp)$ to verify that the platform holds the secret value. Then make a blind signature on $gsk = tsk + hsk$: calculate $a \leftarrow \tilde{g}^y$, $c \leftarrow (a + gpk)^x$. send $cred \leftarrow (a, c)$ as DAA certificate to Host.

5) Host verifies the validity of the certificate, selects two small exponents $e_1, e_2 \leftarrow Z_q$ at random and verifies if $e(e_1 \cdot a, g_2) \cdot e(-e_1 \cdot \tilde{g}, Y) \cdot e(e_2 \cdot c, g_2) \cdot e(-e_2 \cdot (a + gpk), X) = 1$ is valid. If both are valid, the certificate is valid and Host saves the DAA certificate $cred' = (a, \tilde{g}, c, gpk, n)$.

C. *Sign protocol*

The Sign protocol runs between TPM and Host, where a platform with a legitimate certificate performs a signature operation on the data message, and its identity is verified by a trusted third-party server or other trusted platform. Both TPM and Host work together to complete a signature on message *m*. bsn_L can control whether the generated signature is relevant or not. The process is as follows.

1) Host finds the record $(hsk, cred')$ for the Join protocol. Randomly select a $r \leftarrow Z_q$ and blind the DAA certificate $cred' = (a, b, c, gpk, n)$: $a' \leftarrow a^r$, $b' \leftarrow b^r$, $c' \leftarrow c^r$, $gpk' \leftarrow gpk^r$.

2) Host and TPM jointly calculate the nym value for signature relevance detection and a zero-knowledge proof of the gsk secret value. Then enter the parameter $(hsk, tpk', (0 \| n), (1 \| n), m, ("sign", SRL))$, and the output (nym, π'_{cred}) is obtained via the Prove protocol. Where the fourth parameter is set to $\perp$ if the signature is not required to provide correlation.

3) Host calculates and generates the final signature $\pi_{cred} \leftarrow (a', b', c', gpk', nym, \pi'_{cred})$.

D. Verify protocol

The Verify protocol is a verification algorithm used by protocol entities in the environment to verify the legitimacy of a message signature on a (m, π_{cred}) to authenticate the identity of the platform, and every message delivery by the device is subject to a strict signature verification step. The process is as follows.

1) The verifying party queries the list of secret values of the breached platform $Roughlist$. $\forall gsk \in Roughlist$, if there is $gpk' = b'^{gsk}$, the fake platform attack is detected and this attestation is abandoned. Otherwise, go to the next step.

2) The verifier verifies the validity of the DAA certificate. Randomly select two small exponents $e_1, e_2 \leftarrow Z_q$ and verify whether the equation $e(e_1 \cdot a', g_2) \cdot e(-e_1 \cdot b', Y) \cdot e(e_2 \cdot c', g_2) \cdot e(-e_2 \cdot (a' + gpk'), X) = 1$ is valid. If it does not hold, the certification is abandoned. Otherwise, go to the next step.

3) Enter the needed parameter $(\pi_{cred}, nym, (0 \| n), (1 \| n), m, ("sign", SRL))$ to verify that the party is authenticating the validity of the signature via $VerSPK$. If the output is 1, then the attestation is passed. If not, the attestation fails.

E. Mimic defense mechanism

Because the core idea of the DAA protocol is to protect the anonymity of the platform and prevent the included protocol parties from being spoofed by the fake platform, existing DAA schemes always default to the Verifier's identity being legitimate, and the Verifier does not need to self-certify its identity. An attacker-controlled Verifier can receive the platform's signature information without restriction and then return the verified result without raising the platform's suspicion. However, in an M2M communication system, user data is the most important information for an attacker to focus on, and if the platform's signature is not verified and published, there is a high probability that it can be obtained and attacked. Therefore, in this paper, we add mimic security mechanisms to the DAA scheme and take the approach of verifying the platform by setting up multiple heterogeneous executors [12] in a domain as part of the Verifier.

Fig. 4 shows a typical dynamic heterogeneous redundancy architecture for a mimic defense system [13]. When there is a message input, it is transmitted to each heterogeneous executor in the heterogeneous pool through an input agent. All heterogeneous executors process the message and transmit the result to the multimode adjudication module, if the result is consistent, the output will be output; if not, an abnormality can be identified in the message output of an executor, thus

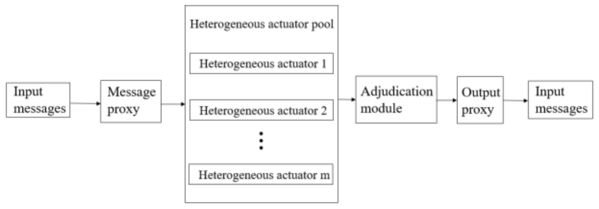

Fig. 4 Mimic defense system dynamic redundancy architecture diagram

realizing the security defense of the system. The process is as follows.

1) Enter a message to start the mimic defense mechanism and perform a verification operation on the signature generated by the Host.

2) Enter the proxy, pass in the parameters required for the Verify protocol, including a', b', c', gpk', e_1, e_2, g_2, the public key pair (X, Y), and the parameter $(\pi_{cred}, nym, (0 \| n), (1 \| n), m, ("sign", SRL))$, and convert them to the form that can be recognized by the heterogeneous executor. If a heterogeneous executor compiled in C, convert these parameters to machine code that can be recognized in that language. Because the subsequent selection of the heterogeneous executor is random, these parameters need to be recognizable to all heterogeneous executors.

3) Grouping of heterogeneous executors, the random number module generates a random number u as the number of heterogeneous executors in each group, and randomly assigns all heterogeneous executors into the group with the number u. The random number module generates a random number u as the number of heterogeneous executors in each group.

4) The random number module generates a number of random numbers $x_1, x_2, ..., x_m$, which are encrypted and distributed to each group of heterogeneous executors, and each group of heterogeneous executors encrypts the encrypted random numbers as the number of each group. If the resulting random number is x_1, first the random number module encrypts x_1 with its own key to get $E_1(x_1)$, and then the heterogeneous actuator group encrypts it with its own key to get $E_2(E_1(x_1))$, the above process is completed inside the Verifier and does not involve other communication objects. The above process is shown in Fig.5.

5) At each verification of the legitimacy of the access device signature, one or more heterogeneous groups of executors are randomly selected to perform the steps in the

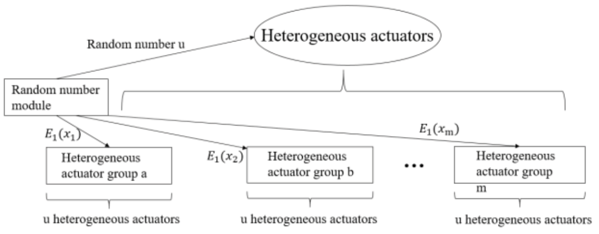
Random number u
Heterogeneous actuators
Random number module
$E_1(x_1)$
$E_1(x_2)$
$E_1(x_m)$
Heterogeneous actuator group a
Heterogeneous actuator group b
Heterogeneous actuator group m
u heterogeneous actuators
u heterogeneous actuators
u heterogeneous actuators

Fig. 5 Heterogeneous actuator grouping method diagram

Verify protocol on the signature to determine whether the device signature is legitimate. Since the number of each heterogeneous executor group is encrypted $E_2(E_1(x_m))$ and the identity information is unknown to the selecting party, only random selection can be made when selecting a heterogeneous executor group.

6) The selected heterogeneous executors execute the Verify protocol, judge the input parameters, and verify whether the signature is legal or not, here the message proxy method is used, if the verification passes the output is 1, otherwise 0.

7) Send the output results of the heterogeneous actuators to the ruling module. If the output results of several heterogeneous actuators are consistent, the ruling module outputs the corresponding ruling results; otherwise, the system is judged to have been attacked. If the output results of the heterogeneous executors are inconsistent when the adjudication module is working, it can also determine that the heterogeneous executors have been attacked in time, and then the system can be repaired in time.

IV. THEORETICAL ANALYSIS

A. Security Analysis

The security analysis of the Mimic-DAA scheme mainly includes four aspects: anti-malicious TPM spoofing, platform anonymity, signature non-forgeability and system security.

1) Anti-Malicious TPM Spoofing: The DAA scheme verifies the legitimacy of TPMs in the Join protocol in order to resist spoofing attacks by malicious TPMs. In the Join protocol, Host submits its own platform information (i,gsk,I) to a trusted third party T to confirm the application for a DAA certificate, T queries the $Roughlist$ to confirm whether the platform has been breached, and communicates the results to Issuer. It is up to Issuer to decide whether or not to issue a certificate, and if Issuer allows it, T saves the platform information to the Members list and tells Host the result. In signature verification, Host sends the information (m,i,gsk,I) to T, and T looks up whether there is a record of the platform information in the Members list, and then determines whether the TPM is illegally forged. Only if there is a record of this platform information in the Members list, T will get the base name bsn information from Host.

2) Platform anonymity: Anonymity can only be guaranteed by an honest platform consisting of an honest TPM and an honest Host. On the one hand, the Sign process uses a new random secret value gsk to generate new signatures corresponding to the base name bsn each time, so that the signatures generated each time are independent from the previous platform, and it is impossible to distinguish the correspondence of signatures generated by different base names bsn from the distribution, so that the anonymity of the platform can be guaranteed. On the other hand, because the platform blinds the certificate, Issuer can only look up the identity of the platform if it gets the original certificate, and it is mathematically impossible to derive the original certificate in turn from the blinded certificate, so the true identity of the platform cannot be obtained even if Issuer and Verifier collude, and the anonymity of the platform is guaranteed when Issuer and Verifier collude.

3) Signature non-forgery: Discussion of non-falsifiability presupposes that by default Issuer is honest. An illegal attacker cannot generate a signature generated by (m,bsn) out of thin air, and every signature that passes Verifier detection must correspond to the secret value of the TPM's gsk , in the case of a legitimate TPM, all legitimate signatures not generated by that TPM cannot be verified by Verifier, so the signatures are non-forgeable.

4) System security: The mimic defense mechanism adopted in this scheme transforms the idea of active defense into a systematic defense mechanism, which dynamically schedules the heterogeneous pool of redundant executing bodies and uses different scheduling strategies to constantly change the components and composition of the target object, increasing the time and space complexity of the system. So an attacker is faced with a completely new system structure each time, greater resources are required to attack the entire system. The number of heterogeneous actuators and objects per scheduling becomes unpredictable in the Mimic-DAA scheme due to the presence of a random number generator. From an active defense perspective, the difficulty of inferring the internal composition of the system from behavior is determined by the difference between the heterogeneous executor of one dispatch and the previous one. In Mimic-DAA, the correlation between each scheduled heterogeneous group of executors and the previous heterogeneous group of executors is also not known. So while the maximum variance between objects per scheduling cannot be satisfied, the correlation between groups of isomers per scheduling cannot be known, so the variance between the two schedules becomes more random and instead makes it more difficult to attack.

B. Validity Analysis

1) Optimization of the DAA scheme: In the execution of attestation protocols, the main computations are focused on the exponential operations and bilinear mappings, so the analysis of efficiency should be measured by the amount of computation required to perform these two operations by the various entities involved in the protocol. The efficiency of the Mimic-DAA scheme is analyzed by comparing it with the schemes already proposed in the literature, as shown in Tables 2 and 3, both of which contain only the amount of computation required to perform each protocol in Mimic-DAA.

In these tables P denotes a pairing operation, G_1 denotes a power operation in G_1 , G_1^2 、 G_2^2 denote multiple power operations.

Table 2 Comparison of computational cost between parties in the Join phase

Stage	Scheme	TPM	Host	Issuer
Join	SC-DAA[14]	$3G_1$	$1G_1 + 2P$	$2G_1 + nG_1 + 1G_1^2$
	I-DAA[15]	$1G_1$	$2P$	$nG_1 + 1G_1^2$
	TMZ-DAA[16]	$4G_1$	0	$2G_1^2$
	Mimic-DAA	$2G_1$	$2G_1 + 1P^4$	$1G_1 + 1G_1^2$

Table 3 Comparison of computational cost between parties in the Sign/Verify phase

Stage	Scheme	TPM	Host	Verifier
Sign/ Verify	SC-DAA	$3G_1$	$3G_1 + G_T + 1P$	$2G_1 + 1G_1^2 + 2P + nG_1$
	I-DAA	$1G_1 + 1G_T$	$4G_1$	$1G_1^2 + 1G_2^2 + 4P + nG_1$
	TMZ-DAA	$3G_1$	$2G_1$	$2G_1^2 + nG_1$
	Mimic-DAA	$1G_1$	$6G_1$	$i(1G_1^2 + 1P^4 + nG_1)$

Based on the above table, it can be seen that the computations of the other protocol parties are optimal in all scenario comparisons except for Host. In Verifier, nG_1 is the computation that substitutes sequentially for the publicly available counterfeit platform *gsk* in *Roughlist* , $1P^4$ is the computation that verifies the legitimacy of the certificate, $1G_1^2$ is the computation that verifies the identity of the platform using zero-knowledge techniques, and the coefficient *i* on Mimic-DAA's Verifier computation is the number of heterogeneous executables selected at a time. The core of the operational efficiency of the DAA protocol lies in the TPM, and because of its limited computational power, optimization schemes regarding efficiency are focused on reducing the TPM computation. The number of Verifier calculations is actually increased in this scheme, but the behavior becomes unpredictable due to the presence of the mimicry mechanism, where the Verifier calculations are different each time the attestation process takes place.

2) Optimization of the mimic defense component: The idea of mimic defense technology is to provide a robust service of "high reliability, high trust, and high availability" for hardware and software through a generalized robust control architecture and other supporting mechanisms.

On the one hand, the introduction of mimic defense modules in the DAA protocol comes at the expense of some of the computational efficiency. But for M2M communication networks, only the TPM module has the problem of low computational power, so setting up multiple heterogeneous actuators in a domain to form Verifier has no change in the computational requirements of the TPM, and now mainframe servers are set up to be able to satisfy the computational power required by multiple heterogeneous actuators. During a complete protocol, only the heterogeneous executors in Verifier are iterated and judged, which is a gain in efficiency for the entire M2M network.

On the other hand, when mimic defense mechanism is introduced, the grouping operation of heterogeneous executors is performed so that the random number module does not need to generate too many random numbers, which saves the resource consumption and improves the computing efficiency. Also, the number and members of heterogeneous executors in the group are different after each grouping, which prevents the previous behavior of the heterogeneous executors from being recorded and improves the security of the system.

CONCLUSION

This paper proposes an anonymous attestation method for M2M networks based on the mimic defense principle, applied in the scenario of device-to-device data communication within an M2M communication system. Firstly, the existing DAA scheme is optimized, and Verifier verifies the signature of the DAA certificate and the legitimacy of the platform after the platform is blinded. Also, because the Verifier identity is always legitimate by default, in order to prevent hijacking of M2M device access, a mimic defense mechanism is added to the Verifier side, using multiple dynamically redundant heterogeneous executors to form the attestation side, which can effectively defend against illegal third-party attacks. If the output of the heterogeneous executor is inconsistent when the adjudication module is working, it can also determine whether the heterogeneous executor is under attack and take effective defensive measures in time. The computational power requirements of the proposed scheme in this paper are high for mimic defense mechanisms, so subsequent work should continue to optimize the algorithm for mimic adjudication to conserve resources and expand the capacity of simultaneous access devices.

REFERENCES

[1] Y. Cao, T. Jiang and Z. Han, “A survey of emerging M2M systems: Context task and objective,” IEEE Internet of Things Journal, vol. 3, no. 6, pp. 1246-1258, December 2016.

[2] B. S. Manoj, A. Chakraborty and R. Singh, Complex Networks: A Networking and Signal Processing Perspective, New Jersey, USA:Prentice Hall PTR, February 2018.

[3] I. Cha, Y. Shah, A. U. Schmidt, A. Leicher, M. V. Meyerstein, “Trust in M2M communication,” IEEE Vehicular Technology Magazine, vol. 4, no. 3, pp. 69-75, 2009.

[4] Trusted Computing Platform Alliance. Trusted computing platform alliance (tcpa) main specification version 1.1b, Beaverton, USA: Trusted Computing Group, 2001.

[5] J. X. Wu, “Cyberspace Mimic Defense.” Technical report, National Digital Switching System Engineering & Technological R&D Center, 2015.

[6] J. X. Wu, “Meaning and Vision of Mimic Computing and Mimic Security Defense,” Journal of Telecommunications Science, vol.30, no. 7, pp. 1-7 (in Chinese), 2014.

[7] E.F Brickell, J. Camenisch, L. Chen, “Direct anonymous attestation,” Proceedings of the 11th ACM conference on Computer and communications security. USA, pp. 132-145, 2004.

[8] J. Camenisch, M Drijvers, and A. Lehmann, “Anonymous attestation using the strong diffie hellman assumption revisited,” The 9th International Conference on Trust and Trustworthy Computing. Germany, pp.1–20, 2016.

[9] L. Chen, R, Urian, “DAA-A: Direct anonymous attestation with attributes,” International Conference on Trust and Trustworthy Computing. Germany, pp. 228-245, 2015.

[10] J. Camenisch, M. Drijvers, A. Lehmann, “Universally composable direct anonymous attestation,” Public-Key Cryptography--PKC 2016. Germany, pp. 234-264, 2016.

[11] L. Xi, K. Yang, Z. Zhang, D. Feng, “DAA-related APIs in TPM 2.0 revisited,” International Conference on Trust and Trustworthy Computing. Germany, pp.1-18, 2014.

[12] J. Voas, A. Ghosh, F. Charron, L. Kassab, “Reducing uncertainty about common-mode failures,” IEEE International Symposium on Software Reliability Engineering. USA, pp.308-319,1997.

[13] B. Stanczyk, A. Peer, M. Buss, “Development of a high-performance haptic telemanipulation system with dissimilar kinematics,” Advanced robotics, vol.20, no. 11, pp. 1303-1320, 2006.

[14] C. Song, Y. Q. Sun,W. P. Peng, “Improved Direct Anonymous Attestation Scheme. Journal of Beijing University of Posts and Telecommunications,” vol.34, no. 3, pp. 62-65, 2011.

[15] L. Chen, Y. He, L. Wang, “Improved direct anonymous attestation scheme in M2M network system,” Journal of Southeast University (Natural Science Edition), vol.42, no. 4, pp. 604-608, 2012.

[16] L. Chen, D Page, N P Smart. “On the design and implementation of an efficient DAA scheme.” 9th IFIP International Conference on Smart Card Research and Advanced Application. Germany: Springer,pp: 223−237, 2010.