Attribute Verifier for Internet of Things

Mohammed B. M. Kamel*†‡, Yuping Yan*, Peter Ligeti*, Christoph Reich†
*Department of Computer Algebra, Eotvos Lorand University, Budapest, Hungary
†Department of Computer Science, Furtwangen University, Furtwangen, Germany
‡Department of Computer Science, University of Kufa, Najaf, Iraq

2022 32nd International Telecommunication Networks and Applications Conference (ITNAC) | 978-1-6654-7103-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITNAC55475.2022.9998348

***Abstract*—Identity management, authentication, and attribute verification are among the main concerns in many Internet of Things (IoT) applications. Considering the privacy concerns, attribute verification became more important in many applications. Many of the proposed models in this field suffer from privacy and scalability issues as they depend on a centralized entity. In this paper, we proposed a decentralized attribute verifier based on a challenge-response approach. To address various IoT attribute verification requirements, the proposed model provides two modes of attribute verification, namely *1-out-of-n verification* and *n-out-of-n verification* modes, in which the participants can prove the possession of one or all of the given target attributes.**

***Index Terms*—attribute-based encryption, attribute verification, zero-knowledge proof**

I. Introduction

With the wide spread use of the distributed applications and web services and a significant increase of public privacy awareness, a major privacy challenge is how to authenticate the various members of a system and their attributes yet manage and protect the privacy of individual digital identities. Considering the privacy concerns, attribute verification became more important in many applications. Assume an IoT auditing system [1] where involving in the auditing process requires that an entity having one of a predefined set of attributes. Considering the privacy requirement, an auditor in the system wants to only prove the fact that it has one of the required defined attributes to be able to join the auditing process, without actually revealing any other information about which exact attributes it has or its other attributes that are not part of the auditing requirement. In other situations, having all of the predefined set of attributes might be required to involve in the auditing process. Similarly, the auditor wants to only prove the fact that it has all of the required defined attributes without revealing any other information about its other attributes that are not part of the auditing requirement. In this paper, we proposed an attribute verifier, that is to the best of our knowledge the first fully decentralized attribute verifier model. The main contributions of this model are the following:

- Two attribute verification modes, namely *1-out-of-n* and *n-out-of-n* verification modes have been proposed which meet the requirements of practical usage.
- A fully decentralized approach for attribute verification.

II. Related Works

From the attribute revocation point of view, Norio et al. [2] outlined a solution for efficient anonymous credential system based on strong Diffie-Hellman assumption. The security properties were proved in the aspects of perfect anonymity-unlinkability and computational unforgeability. Steuer et al. [3] introduced the identity attribute verification scheme in Windows CardSpace. It can compromise the semi-honest central identity manager and does not require information storage by third parties. Meanwhile, the researchers addressed the problem of linkability of digital identities in a privacy preserving approach.

Closely to our work, Guo et al. [4] proposed an attribute proof scheme for smart devices. Their proposed model was based on random oracle pairing-based anonymous credential systems. It efficiently constructs anonymous credentials with cryptographical building blocks. Although the discussed results have important features, their implementation in some IoT applications might be problematic, and can be implemented in specific applications where the existence of a single trusted entity, e.g. the root authority in [5] or [4], is allowed.

III. Attribute Verification

In this section, the proposed attribute verifier model is explained in detail. There are three types of involved participants, the *prover*, the *issuer*, and the *verifier*.

- **Prover**: Proofs the ownership of required attributes through responding to a challenge.
- **Issuer**: Issues the attribute proof to a prover.
- **Verifier**: Verifies the ownership of a prover's attribute through a challenge.

At any given time, any node can join the system as a prover. An organization responsible to handling and managing an attribute i (i.e. the attribute authority i) can join the system and become an independent *issuer*. After confirming that a user (later, the prover) has the claimed attribute, the corresponding issuer sends a secret key of the confirmed attribute to the user. The key exchange confidentiality between the issuer and the prover is out of the scope of this paper. However, protocols such as proposed in [6] can be integrated in the proposed model to achieve the key exchange confidentiality. For verifiers in our proposed model, there are modes: *1-out-of-n* and *n-out-of-n* verification mode. In *1-out-of-n* verification mode, the verifier

This research was partially supported by Project no. TKP2021-NVA-29 has been implemented with the support provided by the Ministry of Innovation and Technology from the National Research, Development and Innovation Fund.

can ask the prover to prove having one of the given n attributes, in which the prover by solving the challenge can can prove the possession of one of the attributes. In contrast, in *n-out-of-n* verification mode, the verifier checks that the prover has all of the given attributes.

We assume that the provers in the proposed model are independent nodes, and each of them has some verifiable attributes, namely the Prover Attribute Vector (PAV). The attributes in PAV vector of each prover may belong to different issuers, and their corresponding secret keys are issued by corresponding issuers without any cooperation between other issuers in the system. PAV by default is private, and only known by the prover node itself. This is a requirement in some applications, such as resource discovery protocols [7]. However, in some applications [8], [9] with specific requirements this vector might be publicly published. We need that the proposed model be sound, and prevents successful verification of malicious provers. On the other hand, considering that PAV is private, the system needs to satisfy the unlinkability and untraceability properties that is required for such systems.

Definition 1 (Soundness): Every Probabilistic Polynomial-Time (PPT) prover can provide a verifiable PAV for forged attributes with negligible probability only.

Definition 2 (Unlinkability): Every PPT verifier can learn whether a pair of attribute verifier tokens is done by the same prover, even by combing attribute verifier tokens from other verifiers with negligible probability only.

Definition 3 (Untraceability): Every PPT issuer can trace the usage of an issued attribute to a prover in the attribute verifier model with negligible probability only.

A. Proposed Model

The proposed model utilizes the Decentralized Attribute-based Encryption (DABE) [10] variant of Attribute-based Encryption (ABE) [11]. Prior to running the proposed model, a global parameter should be generated. It includes two cyclic groups G, G_T, a generator g in G, a bilinear mapping $e : G \times G \rightarrow G_T$ and a hash function $\mathcal{H} : \{0,1\}^* \rightarrow G$ that maps a given identifier to an element in G, a hash function $H : \{0,1\}^* \rightarrow \{0,1\}^d$ that maps any input to a d-bit digest. The order of G is a prime p and the operations are running over some finite fields of p elements, hence every computation are reduced mod p.

$$G, G_T, g \in G, e(.,.), \mathcal{H}(.), H(.)$$

An issuer joins the system by choosing two random private keys $\alpha_i, \beta_i \in \mathbb{Z}_p$ that will be kept private by the issuer itself, and computing the pair of $e(g,g)^{\alpha_i}, g^{\beta_i}$ as its public key. A prover in the system gets the secret keys $sk_{(*,u)}$ of its set of attributes by contacting the relevant issuers. An issuer of an attribute $i$generates the user's corresponding secret key using

$$sk_{(i,u)} = g^{\alpha_i} \mathcal{H}(I_u)^{\beta_i}$$

The verifier defines the set of target attributes $T_v = \{t_0, t_1, \ldots, t_n\}$ for verification. It also randomly generate a challenge key $R \in G_T$. The verifier prepares a challenge by first hashing the challenge key R that will be used as the key to the symmetrically encrypted challenge. The encrypted challenge will be encrypted using the key $k = H(R)$. It includes a nonce $r \in \mathbb{Z}_p$, the timestamp ts, and the public key of the verifier PK_r to be used later to secure the returned response, where $||$ defines the concatenation.

$$challenge = Enc_{H(R)}(r||ts||PK_v) \tag{1}$$

The verifier then generates a random number $s \in \mathbb{Z}_p$, and converts the access policy Γ to the equivalent linear secret sharing scheme (LSSS) matrix $\mathcal{M}(\Gamma)$. The access policy can be set in one of the following two verification modes:

a) 1-out-of-n Verification: The access policy will be set as a Boolean formula with the set of target attributes T_v, joined using an OR operator.

b) n-out-of-n Verification: The access policy will be set as a Boolean formula with the set of target attributes T_v, joined using an AND operator.

The verifier then gets the public keys (pair of $e(g,g)^{\alpha_i}$ and g^{β_i}) of the issuers based on the used attributes in the target attributes T_v. Based on the number of columns in the LSSS matrix, two vectors γ and ω will be generated where their first elements are set to s and 0, respectively, and the rest elements are chosen randomly from $\mathbb{Z}_p$.

The randomly generated challenge key R in both verification modes will be encrypted using DABE [10]. If the verifier has not the required resources to perform the encryption, it can use lightweight protocols such as SDABE [12], OEABE [13], and ODABE [14] to produce the encrypted challenge. To perform the DABE encryption, the verifier and based on the number of rows in LSSS matrix $\mathcal{M}(\Gamma)$ generates three parameters r_i, γ_i, and ω_i for each of the attributes in the set of target attributes T_v. Parameter r_i is a random value that is chosen from $\mathbb{Z}_p$, and γ_i and ω_i are computed using equation (2), where $\mathcal{M}(\Gamma)_i$ denotes the i^{th} row in $\mathcal{M}(\Gamma)$.

$$\gamma_i = \mathcal{M}(\Gamma)_i \gamma, \quad \omega_i = \mathcal{M}(\Gamma)_i \omega \tag{2}$$

The challenge key R will encrypted using equation 3. Additionally for each attribute i in the set of target attributes T_v, three components C_{i_1}, C_{i_2} and C_{i_3} will be computed using equation (4). These components will be used by the prover to get the encrypted challenge key R, i.e. C_0

$$C_0 = Re(g,g)^s \tag{3}$$

$$C_{i_1} = e(g,g)^{\gamma_i} e(g,g)^{\alpha_i r_i}, C_{i_2} = g^{r_i}, C_{i_3} = g^{\beta_i r_i} g^{\omega_i} \tag{4}$$

The prover can proof the claimed attributes if the defined Γ taking its attributes returns true. If the *1-out-of-n* verification mode is used, then having any of the defined attributes in the set of target attributes will satisfy the Boolean formula. On the other hand, if the *n-out-of-n* verification mode is used, the the prover needs to have the secret keys of all attributes in the set of target attributes T_v to be able to get the challenge key, i.e. R challenge key. In order to decrypt the C_0, the prover uses its secret key $sk_{(i,u)}$ and the parameter $C_i = (C_{i_1}, C_{i_2}, C_{i_3})$ of an attribute i in the set of target attributes T_v to compute an intermediate value for attribute i as in equation (5).

$$\frac{C_{i_1}.e(\mathcal{H}(I_u), C_{i_3})}{e(sk_{(i,u)}, C_{i_2})} = e(g,g)^{\gamma_i} e(\mathcal{H}(I_u), g)^{\omega_i} \tag{5}$$

A single computed intermediate value of any attribute i in T_v (in case of *1-out-of-n* verification mode), or all computed intermediate values of all attributes in T_v (in case of *n-out-of-n* verification mode) by equation (5) will be used in equation (6) to compute $e(g,g)^s$:

$$e(g,g)^s = \begin{cases} e(g,g)^{\gamma_i} & \textit{1-out-of-n mode} \\ \prod_{i=1}^{|\Gamma|} e(g,g)^{\gamma_i} e(\mathcal{H}(I_u), g)^{\omega_i} & \textit{n-out-of-n mode} \end{cases} \tag{6}$$

The challenge key R is recovered from C_0 using:

$$R = \frac{C_0}{e(g,g)^s} \tag{7}$$

By recovering the challenge key R, the prover is able to decrypt the challenge in 1 using 8.

$$Dec_{H(R)}(challenge) = r||ts||PK_v \tag{8}$$

Finally, by having the nonce r and the time-stamp, the prover sends back the response encrypted with the public key of the verifier PK_v.

B. Analysis

The proposed model is sound such that a challenge create by a verifier in the system can only be solved by a prover that does not have the defined (to be verified) attributes with negligible probability only, as a consequence of the security of DABE. The unlinkability and untraceability properties are also achieved as a direct consequence of the security and function of underlying cryptographic primitive DABE in our attribute verifier model. Since the challenge and response messages have no information about the identity of the prover (only the fact the a prover has the specified attribute is verified by the verifier), then the verifier by combining the attribute verifier tokens can not learn whether they have been done by the same prover or not. On the other hand, since the issuer is involved only in the process of issuing the attribute for the prover and not in the process of attribute verification through challenge/response approach, then the issuer can not learn the future usage of the issued attributes for the prover.

IV. Conclusion

The centralized models for attribute verification helps verifying the attributes of the nodes in an entity with a high computation capability, and as a result, less complexity. However, the centralized entity in the centralized solutions might turn into a single point of failure and attack, which, if fails, the overall attribute verification system will stop. In this paper, we proposed a decentralized attribute verifier with *1-out-of-n* and *n-out-of-n* verification mode. The proposed attribute verifier is able to verify the attribute in a decentralized and zero-knowledge approach. Since heavy computations through expensive exponentiation are involved, further analysis are required to study the efficiency, complexity and security of the proposed attribute verifier.

References

[1] S. Mirzamohammadi, J. A. Chen, A. A. Sani, S. Mehrotra, and G. Tsudik, "Ditio: Trustworthy auditing of sensor activities in mobile & iot devices," in *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, 2017, pp. 1–14.

[2] N. Akagi, Y. Manabe, and T. Okamoto, "An efficient anonymous credential system," in *International Conference on Financial Cryptography and Data Security*. Springer, 2008, pp. 272–286.

[3] K. Steuer Jr, R. Fernando, and E. Bertino, "Privacy preserving identity attribute verification in windows cardspace," in *Proceedings of the 6th ACM workshop on Digital identity management*, 2010, pp. 13–16.

[4] N. Guo, T. Gao, and H. Park, "Random oracle-based anonymous credential system for efficient attributes proof on smart devices," *Soft Computing*, vol. 20, no. 5, pp. 1781–1791, 2016.

[5] J. Blömer and J. Bobolz, "Delegatable attribute-based anonymous credentials from dynamically malleable signatures," in *International Conference on Applied Cryptography and Network Security*. Springer, 2018, pp. 221–239.

[6] B. Bat-Erdene, Y. Yan, M. B. Kamel, and P. Ligeti, "Security verification of key exchange in ciphertext-policy attribute based encryption," in *7th International Conference on Signal and Image Processing*. IEEE, 2022, pp. 377–381.

[7] M. B. Kamel, Y. Yan, P. Ligeti, and C. Reich, "Attred: Attribute based resource discovery for iot," *Sensors*, vol. 21, no. 14, p. 4721, 2021.

[8] M. B. Kamel, K. Wallis, P. Ligeti, and C. Reich, "Distributed data validation network in iot: a decentralized validator selection model," in *Proceedings of the 10th International Conference on the Internet of Things*, 2020, pp. 1–8.

[9] M. B. Kamel, P. Ligeti, and C. Reich, "D3vn: Decentralized abe-based distributed data validation network," in *Proceedings of Seventh International Congress on Information and Communication Technology*. Springer, 2022.

[10] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2011, pp. 568–588.

[11] Y. Yan, M. B. Kamel, and P. Ligeti, "Attribute-based encryption in cloud computing environment," in *2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. IEEE, 2020, pp. 63–68.

[12] M. B. Kamel, P. Ligeti, and C. Reich, "Sdabe: Efficient encryption in decentralized cp-abe using secret sharing," in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 2022, pp. 1–6.

[13] K. T. Nguyen, N. Oualha, and M. Laurent, "Securely outsourcing the ciphertext-policy attribute-based encryption," *World Wide Web*, vol. 21, no. 1, pp. 169–183, 2018.

[14] M. B. Kamel, P. Ligeti, and C. Reich, "Odabe: Outsourced decentralized cp-abe in internet of things," in *Applied Cryptography and Network Security Workshops*. Springer, 2022.