A Scalable Model for Secure Multiparty Authentication

Hussain Al-Aqrabi and Richard Hill
Centre for Industrial Analytics
University of Huddersfield
Huddersfield, HD1 3DH, UK
Email: {h.al-aqrabi, r.hill}@hud.ac.uk

Abstract—**Distributed system architectures such as cloud computing or the emergent architectures of the Internet Of Things, present significant challenges for security and privacy. Specifically, in a complex application there is a need to securely delegate access control mechanisms to one or more parties, who in turn can govern methods that enable multiple other parties to be authenticated in relation to the services that they wish to consume. We identify shortcomings in an existing proposal by Xu et al for multiparty authentication and evaluate a novel model from Al-Aqrabi et al that has been designed specifically for complex multiple security realm environments. The adoption of a Session Authority Cloud ensures that resources for authentication requests are scalable, whilst permitting the necessary architectural abstraction for myriad hardware IoT devices such as actuators and sensor networks, etc. In addition, the ability to ensure that session credentials are confirmed with the relevant resource principles means that the essential rigour for multiparty authentication is established.**

Keywords— Cloud computing, distributed systems, security, authentication, trust, multiparty, Internet of Things

I. INTRODUCTION

As both individuals and organisations embrace the benefits of cloud computing infrastructure, more and more data storage and business process services are being transferred or established in clouds [1]. This shift from local to remote infrastructure drastically reduces the effort and expenditure [4] required for system maintenance [2], [3], enabling system users to concentrate on business concerns such as QoS and performance, etc.

The rapid uptake of services delivered via clouds has meant that matters of convenience have overtaken concerns about security and privacy. Distributed systems such as clouds inherently introduce security vulnerabilities that can be accessed remotely if insufficient security measures are in place. Cloud-based systems have specific limitations with regard to security and these are discussed by Al-Aqrabi et al [5].

There still remain enterprises and individuals who are reluctant to adopt cloud infrastructure, and the lack of awareness of secure methods of cloud adoption limits the business advantages that are available to users and enterprises [6], [7].

Choo [9] and Liu [8] both describe the need to ensure that both the providers and consumers of cloud computing have the appropriate mechanisms deployed for security and privacy. As such there has been considerable research activity [12], [13], [16], [17] pertaining to the security of cloud applications and infrastructure [10], [11], [14], [15].

A. Multiparty service delivery and security

As enterprises are beginning to become aware of the power of data collection, analysis, modelling and prediction, they are starting to realise systems that are a more faithful representation of business processes. This means that the underlying digital services must demonstrate both robustness and flexibility to tolerate new and unanticipated business scenarios. As such, the actual process flows may be difficult to predict in some instances, especially if a business offers bespoke services or products to customers, where a transaction may execute once only [28].

As a consequence of this, the eventual application that is delivered is underpinned by a collection of disparate services that are orchestrated at run-time, that may have origins in organisations that are heterogeneous. Each of the host organisations will have adopted security measures that are unique to the enterprise, with the effect that an application composed of multiple services will thus present a number of different security realms.

Each realm typically consists of data that represents a collection of resource principals, that are registered with a trusted principal such as a certificate authority. The principals are governed by a set of security policies that control access to other services and resources within the scope of the application [29]. The certificate authority is deemed to be trustworthy across the application domain and is present to validate users and functions [30].

It is essential that each security realm is authenticated against to ensure that a principal has the appropriate security privileges to consume services marshalled by a security realm. The identity of a principal needs to be confirmed by the correct authentication process of the relevant realm so as to correctly identify and establish who the principal is. During the authentication process, security credentials that were given to the principal by the relevant security realm are used to authenticate it.

In the case of more complex application architectures, such as cloud-based services provision, each cloud may hide multiple instances of other clouds and/or services. It follows that not only will there be numerous authentication mechanisms to keep maintained, but they will have to be invoked dynamically at runtime on demand. If separate authentication processes are established across disparate security realms, there is a potential for a significant increase in authentication workload and the consequential side-effects on network bandwidth and computational cycles.

The scenario where a multiparty session is composed of many two-party sessions is explored by Hada et al [31], who demonstrates that there is a need for a protocol for multiparty session authentication. Thre is an inherent challenge here that it is not always possible for a session participant to establish whether another session participant is actually a member of the multiparty session in progress.

The rest of this article is organised as follows. First, we consider the main challenges for secure authentication in distributed systems infrastructure such as cloud computing, where multiple parties are present. In particular, we shall consider the key obstacles that are presented by environments that are composed of many different parties of varying capabilites such as with the Internet of Things (IoT).

Second, we shall briefly review some existing approaches to managing multiparty authentication, and critically discuss a model

developed by Xu [29]. Third, we propose a distinct model for secure multiparty authentication that addresses shortcomings in current models, and explain how it can be deployed by way of an example. Finally, we describe some concluding remarks.

II. KEY CHALLENGES FOR MULTIPARTY ENVIRONMENTS

Whilst multiparty authentication is a complex challenge in a multi-cloud environment, the complexity increases considerably when we consider the potential proliferation of devices in IoT systems. In such systems, there may be 1:1 mappings between system access devices and the clouds themselves, but there is also the additional potential complication of myriad hardware devices that possess varying degrees of functionality and capability. Such devices can be found in Wireless Sensor Networks for instance, which are often adaptive entities that can embrace the addition or removal of sensor nodes during operation.

If Gartner's prediction is true - "Gartner, Inc. forecasts that 8.4 billion connected things will be in use worldwide in 2017, up 31 percent from 2016, and will reach 20.4 billion by 2020." [33] - then the demand for authentication of devices will be a significant challenge to address, particularly since there will be insufficient capacity to manually authenticate even a fraction of the devices, and therefore some automation will be mandatory.

A fundamental challenge in a complicated environment such as the IoT or multi-clouds is the need to manage and assure the communications that enable the requisite authentication approvals to be enacted [17].

The use of Single Sign On (SSO) has become a convention for users to conveniently access systems that are composed of multiple sub-systems, each of which may be a different application deployment. SSO removes the need for users to enter differing security credentials multiple times, and is enabled by the use of a key exchange mechanism to manage the provision of authentication credentials that have been certified by a named authority [18], [19]. However, the relative simplicity of a mechanism to provide a secure method of key exchange is inadequate for the situation when we need multiple parties to be able to trust each other in a dynamic, heterogeneous environment, and therefore SSO is lacking in this regard.

In the next section we shall discuss existing approaches to multi-party authentication.

III. EXISTING MULTI-PARTY APPROACHES

In a multiparty concept, multiple parties can join or leave a session dynamically. The parties are allowed or removed from the session by a session authority. A simplified drawing showing the concept is presented in the below figure.

In this concept, a session authority controls the authentication of all session participants. Existing session participants can introduce new participants to the session authority for limited transactions. The session authority issues a secret session key to all running session participants.

In practice the session authority communicates with multiple session handlers. Whenever a new participant joins or existing one leaves, the key is refreshed and shared with the active session participants using forward security techniques [24].

The session authority recognises the session participants with the help of participant IDs. A participant leaving the session cannot reuse its participant ID for re-entering.

Similarly, a reused ID will not be assigned to a new participant. The participants join through introduction only and need not share any secret artefacts to gain the session key. However, two participants acting as partners can share private keys using the Diffie-Hellman algorithm.

Prior work related to multiparty identity authentication in multi-cloud environments [29], [24] has described models that utilise more sophisticated methods than password management through SSO. The
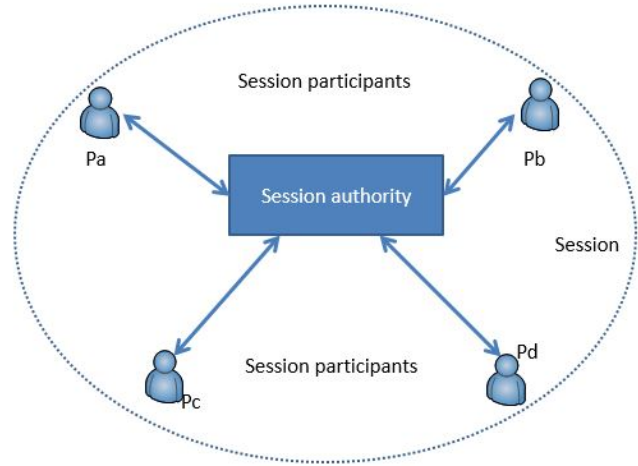

Fig. 1. The multiparty session authentication concept.

concept of direct trust is introduced as part of their mechanism, with [20] introducing group credentials across security realms within inter-cloud communications.

The basic premise is that a new actor (either human or device) should be able to access a system as a result of a secret that is shared amongst a group. This is developed further by [29] and [24] whereby a third party, who is trusted, can discover and establish permissions on behalf of the actor.

In such a scenario, the system can observe that an actor is associated with a group that is trusted, and therefore will assign permissions based on the association. The use of trust relationships opens up the potential for actors to gain access to systems based upon their reputation, which is more akin to human social networks.

At any one time, it is conceivable that there will be many system access requests to complete in a secure and timely manner. This is one aspect of the emerging IoT scenarios that is particularly challenging, in terms of the sheer volume of potential requests that may exist.

The Session Authority (SA) is a role of fundamental importance in that it manages the confirmation and approval of access requests to the system. The SA is preceded by a Multi-Party Session Handler $MPSH$ who formulates a queue of requests for subsequent processing by the (SA).

Since the workload of the SA is likely to vary, and at least will be expected to scale upwards when an application grows by the addition of additional actors and their devices, one approach to deal with the elasticity in demand is to adopt an SA Cloud (SAC) as proposed by [23], [24] and illustrated in Figure 2.

IV. A MODEL FOR MULTIPARTY AUTHENTICATION

In this section, a model for dynamic authentication interactions in a distributed environment (in this example multiple clouds) is shown in Figure 2 [24]. All members of multiple sub-domains may interact within a session and all such sessions are identified by the Session Authority Cloud (SAC). Session keys comprise of the root key of the cloud, a sub-domain key, and the portion identifying the session.

This means that there will be multiple session keys valid for a session, each having a common field for the session, but varying fields for cloud root keys and sub-domain keys. There is no need for any negotiation among the clouds because the session authority cloud is aware of all the clouds and their sub-domains.

The schematic of the robust multiparty model is shown in Figure 2. The ground rules of the proposal are:
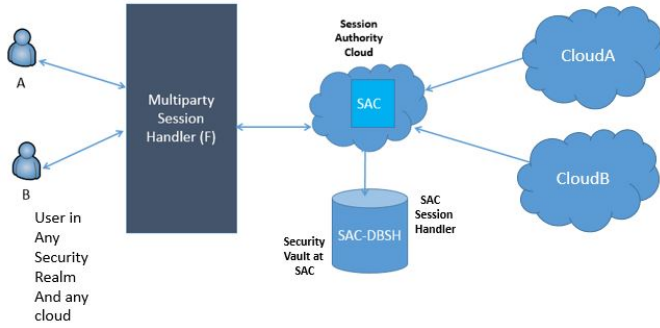
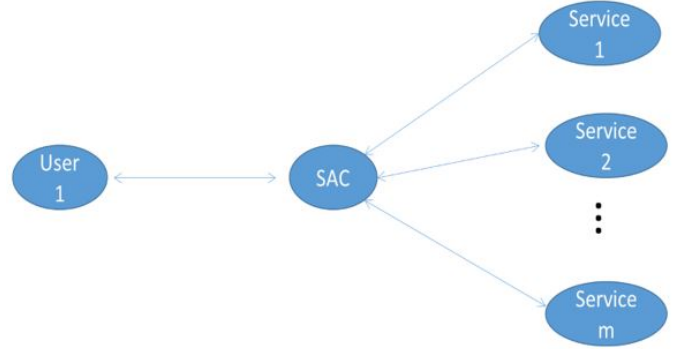Fig. 2. A multiparty session authentication model [24].

Fig. 3. A worst case scenario where either n users and one service, or there are 2 users and 2 services.
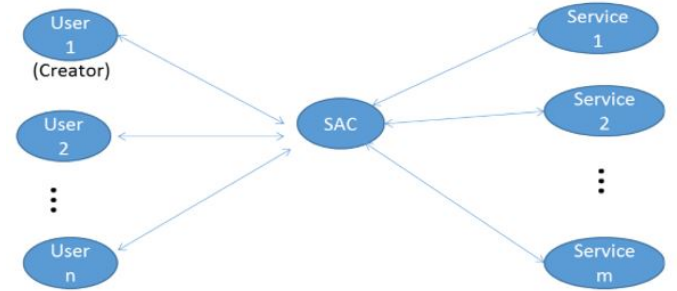

Fig. 4. A best case scenario where there are n session users and m services, where both n and m are much greater than 2.

- Each session participant should be a tenant of at least one cloud in the multi-cloud framework controlled by the session authority cloud.
- If a potential participant is not a cloud member, the introducing participant will have to share credentials with it for joining its own cloud.
- Each session will have multiple valid keys. While the session key field will be common (refreshed on change of no of participants), the cloud root keys and sub-domain (security realm) keys will vary depending upon the membership profiles of the participants.

The session begins with a user having membership in any security realm (cloud C) that the trusted principal recognises. We assume to provide access to some resources such as database objects in clouds A and B if the SAC approves the request forwarded by the principal.

It is also assumed that SAC will not entertain any request not forwarded by the principal. The user requesting access is neither a member of cloud A nor a member of cloud B. In essence, the user is a member of a security realm that is a different cloud (cloud C), which is trusted by the SAC (which means that the third cloud is a member of the $SAC - DB$).

Most importantly, the principal should recognise who is the user because the SAC trusts the principal for accepting the session request. Hence, the only way the user can gain access to database files on clouds A and B is to send a request to the session authority cloud through the session handler F.

The session handler will only forward requests of the trusted principal and hence the requests need to be forwarded through the login of the trusted principal.

The principal A places a request to the session handler to gain access to resources A and B for the user (the user knows their URLs but does not have any access to them).

On the request of the session handler (F), the principal A shares the root and sub domain keys of the user. These keys may be viewed as two packets of a finite size (for example, 1024 Bytes each).

The SAC checks the keys with the help of a database $SAC-DB$ to assist it (may be viewed as a huge security vault having all root and sub-domain keys of the clouds registered with it).

On confirmation from $SAC - DB$, the SAC approves access to database files A and B stored on clouds A and B respectively. It forwards its approval to the SACs session handler ($SAC - SH$). The $SAC - SH$ may be viewed as a separate dynamic database that caches all approvals from the SAC and forwards them to respective clouds for opening the accesses.

Cloud A stores the session ID and key in its registry or cache and then sends a response to SAC. SAC sends a reply for session approval to F. Then, F sends a response for session approval to access the application on cloud A or B.

Typically, the authentication processes occur between SAC and session members (users and services). Figure 3 shows a worst case scenario, where there is only one user and m services in a session. Conversely, worst case scenarios exist when there are either n users and one service, or there are 2 users and 2 services. In each of these cases, the authentication process cannot be simplified and the SAC process offers no significant advantage over direct authentication. In addition, the two-party session technique does not address the issue of different Cross-Realm Authentication, which requires credential conversion and the establishment of authentication paths. On the other hand, Figure 4 shows a best case scenario for the multiparty model. In this case there are n session users and m services, where both n and m are much greater than 2. The benefit is obtained since each user will be able to access all of the m services. Without this model, the authentication processes must be performed within sessions because each user has to be authorised by all of the services that they want to access.

V. Different security realms

If we consider the situation whereby authentication is required across two different security realms, it is necessary to a) convert credentials so that they can be shared, and b) define specific authentication paths between the relevant realms. Existing methods for two-party session authentication are not able to resolve such scenarios.

One approach is to use federated authentication, although the negotiation amongst parties is time intensive, especially when one or more parties requires the authentication path to be amended or augmented to their satisfaction. The complexity of this situation escalates rapidly as more parties are added, either as actors wanting access, or myriad services distributed across differing security realms.

The fact that existing approaches do not comprehensively address the increasingly prevalent situation where multiple actors have a need

to access multiple services, across multiple security realms (such as a multi-cloud environment), means that business models that are based upon the distributed provision of services are potentially hampered by the ability to scale authentication at a less than optimal rate. We foresee this as a significant barrier for secure IoT, as the orchestration of services (latterly microservice architectures [34], [35] held within containers is a natural progression as interest develops for scalable business models and infrastructure.

VI. EVALUATING MULTIPARTY MODELS

We now proceed to examine the multiparty scenario described by Xu [29] in order to understand some of the pertinent challenges faced by multiparty authentication models. The multiparty model of Xu [29] employs a SA entity that oversees the authorisation of sessions against session requests from existing partners. With this arrangement a session can be established by an actor that is able to provide an instance ID; this means that there is no need to identify the resources to be accessed, nor to initiate contact with their principals.

If two session parties, A and B have already communicated and established trust, either A or B can introduce a new party, C. The SA will approve the request merely because A and B are in a current session, irrespective of the means used to access the resources, which may be either legitimate or unauthorised.

A secret session key is provided to C to join the session. If any one of the parties leave the session (for instance A), the secret key will be refreshed such that A cannot rejoin on its own (silently). If A does attempt to join, it will be detected and considered a potential security breach.

Thus, A needs to seek sponsorship of B or C as they are already authenticated in a session, even though this C was introduced to the session by A earlier. Therefore, the model proposed by Xu enables session participants to introduce new parties, a decision that is endorsed by the SA without any conditions.

Furthermore, the SA has no concern what each party is accessing within a session. This is significant in the following situation. If parties A, B, and C are accessing a restricted database, the SA does not seek approval from the owner of the database.

Hence, if C becomes the provider of a session instance ID, as part of an inside attack upon the database, the SA will honour the request as no further checks are made. In fact, the database owner is an invisible security realm for the SA, as it is only concerned with a session S via the session ID and instance ID.

If all three parties were adversarial attackers and were successful at breaching the database be creating a SI and a CI, any request for part D to join the session would be approved by the SA and the actual authority of the database would never be contacted to confirm or deny credentials.

To summarise the observations so far with respect to the multiparty model of Xu [29]:

(a) The SA has no knowledge about resources used in the session and is only concerned with CI and SI;
(b) The SA has no access to knowledge of the principal owners (the actual authority) of resources for a session;
(c) The SA never contacts the principal resource owners as long as an instance key is provided by a party that already has access to the resource;
(d) The SA has no other mediator to assist with authentication checks;
(e) The SA cannot control the addition of access to resources during an executing session.

In contrast, we shall now consider the multiparty model that employs a Session Authority Cloud (SAC) entity [24], illustrated in Figure 6.

The SAC differs from the SA of Xu [29] in that it includes a security vault database ($SAC - DB$) to facilitate authentication checks beyond that of session coordination only. This database
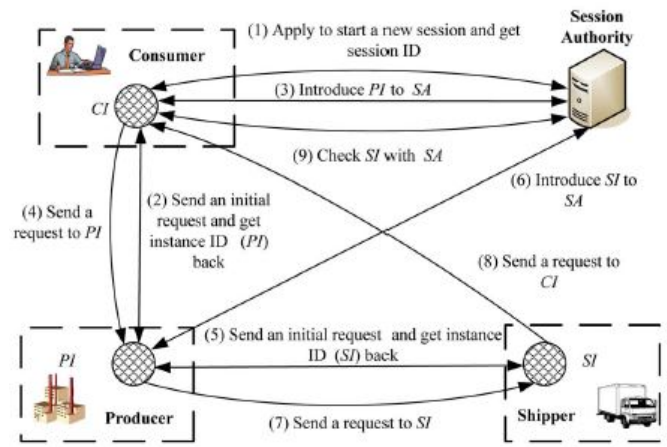

Fig. 5. Multiparty business scenario as described by Xu [29].
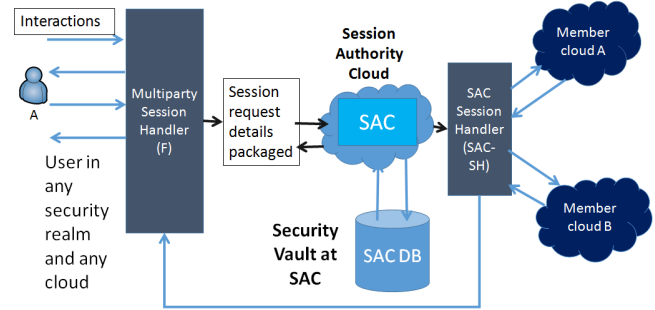

Fig. 6. Multiparty authentication model as described by Al-Aqrabi [24].

provides knowledge of what activities are being performed within a session, which removes the potential vulnerability of the SA having the power to grant session access to a proposer without requiring any resource checks.

The $SAC - DB$, together with the $SAC Session Handler$ ($SAC - SH$), ensure that session participants are authenticated only when all details of the required resources are confirmed, rather than relying only on the collection of session keys as the sole authentication mechanism. If the resource checks fail, the SAC will not allow the session to execute. As such, the SAC contacts the destination resources for confirmation, which is absent from the previous model [29].

The fact that the SA will honour requests for access without further authentication, means that Xu's model cannot protect an application from multiparty adversarial attacks, which are typically initiated from the inside. As the resource owner belongs to a realm that the SA is not concerned with (as it requires only a session and instance ID), one nefarious party can grant access to another.

This could be an inside agent that enables an additional external agent to compromise the application for malicious purposes. We feel that this is a major barrier to the adoption of such a model, particularly for IoT architectures where there is a reasonable expectation that additional devices will be added to the system in the future, that will require authentication through verified credentials.

Approval is requested by the SAC from the resource owner (e.g., database owner), who is a security realm principal and external to the session. Furthermore, the SAC takes individual approvals for the resources accessed and hence, a new resource access request cannot be added during a session that is executing.

For example, if we consider the playing of video content from

embedded links within a database: as soon as an embedded video link is launched, an approval screen with a pop-up directing the users to involve the SAC again, who in turn will seek its access from the principal owner of the videos. The SAC is supported by the $SAC - DB$ and $SAC - SH$ to ensure that the principals of the resources are contacted for appropriate authentication and approval.

Alternatively, an IoT based network may utilise wireless communication to enable wireless sensors and other mobile devices to move into and out of the scope of an application. This might enable a network to utilise opportunistic sensing, or packet transport from IoT appliances such as vehicles. In this situation, it will be necessary to establish a trust relationship that may be transient and limited in its ability to access certain resources within the application. In this case, the SAC again is assisted by the $SAC - DB$ and $SAC - SH$, so that access is provided to the requisite parties in a secure manner.

VII. CONTRIBUTION

A more substantial process of establishing a multiparty session is a key differentiator between the two models. This extra emphasis within the second model [23] ensures that the trust relationship is more secure from the outset.

Access to resources is strictly limited to the approvals by the resource principals of cloud A and cloud B, who are named ownership entities rather than an adversary who succeeds in generating an instance ID. In addition, even if the principals are not in the session, they can monitor it from outside.

Finally, we consider the more demanding scenario of large collections of IoT devices being present as part of emergent system architectures, whose fundamental capabilities have to be based upon secure scalability. The SA entity of Xu's model serves as a constraint in that is cannot function without intervention to generate and share instance IDs.

Thus, the multiparty model proposed in [23] presents two siginificant contributions as follows:

1) The employment of a SAC, together with a $SAC - DB$ and $SAC - SH$ ensures that session authority is granted with full cooperation of the resource principals, which provides more rigour when establishing a trust relationship at the outset;
2) Scalability of authentication requests is catered for through the abstraction of the SAC; hardware devices such as actuators, sensor networks, etc., can be fully hidden behind the SAC and the $SAC - SH$ entities.

Together, contributions (1) and (2) provide a more robust base upon which to consider the security of multiparty application architecture.

VIII. CONCLUSION

In conclusion this article examines approaches to multiparty authentication through the lens of applications that have a future requirement to scale upwards, such as the emergent growth of IoT networks. Such architectures are dynamic, must exhibit elasticity, and also present a multitude of potential security vulnerabilities. Nonetheless, the appeal of distributed hardware that is networked provides a compelling motivation for business and individual users alike to adopt such technologies.

We have selected two proposed models of multiparty authentication and examined their suitability with respect to the IoT scenario described above. We have identified that models which can be utilised for Service Oriented Architectures need to ensure that the authentication controls, policies and protocols need to protect against the inevitable multiparty interactions that will have to be satisfied.

We examine a novel multiparty authentication approach [21], [23] that provides a robust mechanism for marshalling security interactions with distributed infrastructures such as multi-cloud and IoT networks. This model is pertinent when multiple members of heterogeneous security realms have a desire to access a variety of services, under the governance of a trusted principal. The model thus enables authentication dynamically during execution of a multiparty application, whilst ensuring that there is a minimal requirement to convert security credentials as the services are accessed. A major motivation for the adoption of this model is that of authentication simplification between two or more services that are unrelated, before the services are permitted to exchange data. Only when authentication checks have been made with resource principals is data exchange permitted.

We intend to develop this work to identify and address new challenges posed by heterogeneous IoT environments. Key areas of focus are as follows:

- Developing proofs of secure authentication protocols for multiparty scenarios;
- Using hardware-in-the-loop to deploy IoT networks, enabling authentication overheads to be evaluated for a variety of use cases, particularly in relation to different networking protocols such as LoraWAN and Zigbee;
- Monitoring and assessing the impact of energy consumption of the authentication protocols upon constrained hardware resources;
- Investigating methods for the visualisation of authentication and multiparty behavioural signatures to improve resilience towards more sophisticated adversarial attacks;
- Developing mechanisms to automate trust formation through authentication for resource-constrained hardware.
- Exploring the use of SDN to manage network traffic across IoT devices in a multiparty authentication scheme [32].
- Developing resource scheduling algorithms to manage multiparty authentication workload across multiple clouds and IoT hardware [43].

REFERENCES

[1] R. Hill, L. Hirsch, P. Lake, S. Moshiri. Guide to Cloud Computing. Springer-Verlag London. ISBN 978-1-4471-4603-2.

[2] A. A. Albeshri and W. Caelli. Mutual protection in a cloud computing environment, in IEEE 12th International Conference on High Performance Computing and Communications (HPCC), pp641646, 2010.

[3] RN. Calheiros, R. Ranjan, A. Beloglazov, CA. De Rose, R. Buyya. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience 41(1), pp2350, 2011.

[4] G. Bieter. Cloud computing architecture and strategy. IBM Blue Books: 3-4, IBM Research, 2010.

[5] H. Al-Aqrabi, L. Liu, R. Hill, N. Antonopoulos. A Multi-layer Hierarchical Inter-Cloud Connectivity Model for Sequential Packet Inspection of Tenant Sessions Accessing BI as a Service. Proceedings of 6th International Symposium on Cyberspace Safety and Security (CSS) and IEEE 11th International Conference on Embedded Software and Systems (ICESS). France, Paris, March 20-22, IEEE, pp137-144, 2014.

[6] G. Sharma, S. Kalra. Identify based secure authentication scheme based on quantum key distribution for Cloud computing. Peer-to-Peer Networking and applications.(2018) Springer, 11(2), pp220234, 2018.

[7] J. Chen, Y. Wang, X. Wang. On-Demand Security Architecture for Cloud Computing, Computer, 45(7), pp73-78, 2012.

[8] H. Liu, H. Ning, Q. Xiong, LT Yang. Shared authority based privacy-preserving authentication protocol in cloud computing. IEEE Transactions on Parallel and Distributed Systems, 26(1), pp24151, 2015.

[9] K. Choo, O. Rana, M Rajarajan. Cloud Security Engineering: Theory, Practice and Future Research. IEEE Transactions on Cloud Computing, 5(3), July-Sept, 2017.

[10] G. Ateniese, M. Steiner, and G. Tsudik. New multiparty authentication services and key agreement protocols. IEEE Journal on Selected Areas in Communications, 18(4):628639, 2000.

[11] J. Katz, M. Yung: Scalable Protocols for Authenticated Group Key Exchange. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp110125. Springer, Heidelberg, 2003.

[12] D. Thilakanathan, S. Chen, S. Nepal and R. A. Calvo. Secure Data Sharing in the Cloud. In: Security, Privacy and Trust in Cloud Systems, Springer-Verlag Berlin, Heidelberg, 2014.

[13] W. Song, H. Zou, H. Liu, J. Chen. A practical group key management algorithm for cloud data sharing with dynamic group. In: China Communications, 13(6), IEEE Journals and Magazines, 2016.

[14] E. J. Yoon, K. Y. Yoo. New authentication scheme based on a one-way hash function and Diffie Hellman key exchange, 4th International Conference of Cryptology and Network Security, CANS 2005, LNCS vol. 3810, Springer-Verlag, pp147-160, 2005.

[15] Y. Rahulamathavan, R. C.W. Phan, S. Veluru, K. Cumanan, M. Rajarajan. Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud, IEEE Trans. Depend. Secure Comput., vol. 11, no. 5, pp467479, 2014.

[16] P. Kumar Arya, K. Selvamani, S. Kanimozhi, A. Kannan. Data sharing for dynamic group in the cloud environment by using group signature approach. IET Chennai Fourth International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2013), pp449-455, 2013.

[17] A. Celesti, T. Tusa, M. Villari, A. Puliafito. Security and Cloud Computing: InterCloud Identity Management Infrastructure, In 2010 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 28-30 June 2010, Larissa, Greece. IEEE Computer Society, pp263-265, 2010.

[18] R. Sharma, B. Joshi, "H-IBE: Hybrid-identity based encryption approach for cloud security with outsourced revocation", In 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), 3-5 Oct. 2016, Paralakhemundi, India, IEEE Xplore, 2016.

[19] C. Schridde, T. Domemann, E. Juhnke, B. Freisben, M. Smith, "An identity-based security infrastructure for Cloud environments", In 2010 IEEE International Conference on Wireless Communications, Networking and Information Security, 25-27 June 2010, Beijing, China, IEEE Xplore, 2010.

[20] Q. Dai, X. Zhao, X, Q. Xu, H. Jiang, "A New Cross-realm Group Password-based Authenticated Key Exchange Protocol", IEEE Computer Society, p. 856-860, IEEE Xplore, 2011.

[21] H. Al-Aqrabi, L. Liu, R.Hill, N. Antonopoulos. Cloud BI: Future of business intelligence in the Cloud, Journal of Computer System Science, Elsevier, 2014.

[22] H. Al-Aqrabi, L. Liu, R. Hill, L. Cui, J. Li. Faceted Search in Business Intelligence on the Cloud, in Proceedings of GREENCOM-ITHINGS-CPSCOM, pp842-849, 2013.

[23] H. Al-Aqrabi, L. Liu, R. Hill, N. Antonopoulos. Taking the business intelligence to the Clouds. Proceedings of 14th IEEE International Symposium on High Performance Computing and Communications, HPCC 2012, Liverpool, UK, IEEE Computer Society Press, 2012.

[24] H. Al-Aqrabi, R. Hill. Dynamic Multiparty Authentication of Data Analytics Services within Cloud Environments. In 20th IEEE International Conference on High Performance Computing and Communications (HPCC-2018), IEEE Computer Society.

[25] H. Al-Aqrabi, L. Liu, R. Hill, Z. Ding, N. Antonopoulos Business intelligence security on the Clouds: challenges, solutions and future directions. Proceedings of 7th International Symposium on Service-Oriented System Engineering, SOSE 2013, 2528 March 2013, San Francisco, CA, USA, IEEE, 2013.

[26] S. Chaudhuri, U. Dayal, V. Narasayya, An Overview of Business Intelligence Technology. Communications of the ACM , 54(8), pp88-98, 2011.

[27] J. Glaser, J.Stone. Effective use of Business Intelligence. Healthcare Financial Management, 2008, 62(2), pp68-72, ABI/INFORM Global, 2008.

[28] D. Georgakopoulos, M. Hornick, An Overview of Workflow Management: From Process Modelling to Workflow Automation Infrastructure, Distributed and Parallel Database, pp. 119-153, Mar. 2005.

[29] J. Xu, D. Zhang, L. Liu, X. Li, X. Dynamic Authentication for Cross-Realm SOA-Based Business Processes, IEEE Transactions on services computing, 5(1), pp20-32, 2012.

[30] J. D. Clercq. Single Sign-On Architectures, Proc. International Conference, InfraSec 2002, Bristol, UK, pp40-58, 2002.

[31] S. Hada, H. Maruyama. Session Authentication Protocol for Web Services, Proc. Symposium on Application and the Internet, pp158-165, 2002.

[32] C. Baker, A. Anjum, R. Hill, N. Bessis, S. Liaquat Kiani. Improving cloud dtatcentre scalability, agility and performance using OpenFlow. Proceedings of the 4th International Conference on Intelligent Networking and Collaborative Systems (INCoS), IEEE Computer Society, 2012.

[33] Gartner, Inc. Newsroom, "Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016", URL: https://www.gartner.com/newsroom/id/3598917, February 7th 2017.

[34] D. Shadija, M. Rezai, R. Hill. Towards an understanding of microservices, 23rd International Conference on Automation and Computing (ICAC), Huddersfield, UK. IEEE Computer Society, DOI: 10.23919/IConAC.2017.8082018, 2017.

[35] A. Ikram, A. Anjum, R. Hill, N. Antonopoulos, L. Liu, S. Sotiriadis. Approaching the Internet of things (IoT): a modelling, analysis and abstraction framework. Concurrency and Computation: Practice and Experience 27 (8), pp1966-1984, 2015.

[36] M. Burrows, M. Abadi, and R. Needham. A Logic of Authentication, ACM Trans. on Computer Systems, 8(1), pp18-36, 1990.

[37] C. Chen, J. Tu. A Novel Cloud Computing Algorithm of Security and Privacy, Mathematical Problems in Engineering, Hindawi Publishing Corporation, pp1-6, 2013.

[38] J. Li, B. Li, T. Wo, C. Hu, J. Huai, L. Liu, K. Lam. CyberGuarder: A Virtualization Security Assurance Architecture for Green Cloud Computing, Future Generation Computer Systems, Elsevier Science, 28(2), pp379-390, 2012.

[39] S. Pippal, V. Sharma, S. Mishra, D.S. Kushwaha, An Efficient Schema Shared Approach for Cloud based Multitenant Database with Authentication and Authorization Framework, IEEE Computer Society, pp213-218, 2013.

[40] H. Li, Y. Dai, L. Tian, and H. Yang, Identity-Based Authentication for Cloud Computing, Proc. First Int'l Conf. Cloud Computing (CloudCom), Spring, pp157-166, 2009.

[41] H. Li, Y. Dai, B. Yang, Identity-Based Cryptography for Cloud Security, University of Electronic Science and Technology of China and University of Tennessee, USA, https://eprint.iacr.org/2011/169.pdf (Accessed: 2 January 2018), pp1-9, 2011.

[42] B. Qin, H. Wang, Q. Wu, J. Liu, J. Domingo-Ferrer, Simultaneous authentication and secrecy in identity-based data upload to cloud, Cluster Computing, Spring, 16(4), pp845-859, 2013.

[43] S. Sotiriadis, N. Bessis, N. Antonopoulos, R. Hill. Meta-scheduling algorithms for managing inter-clod interoperability. International Journal of High Performance Computing and Networking, 7(2), pp156-172, 2013.