Network and Communication Technologies; Vol. 5, No. 1; 2020
ISSN 1927-064X E-ISSN 1927-0658
Published by Canadian Center of Science and Education

A Secured System for Internet Enabled Host Devices

Aderonke F. Thompson [1], Oghenerukevwe E. Oyinloye [2], Matthew T. David[3] & Boniface K. Alese[2,4]

[1] Cyber Security Department, The Federal University of Technology, Akure, Nigeria

[2] C omputer Science Department, Ekiti State University, Ado-Ekiti, Nigeria

[3] Computer Science Department, The Federal University of Technology, Akure, Nigeria

[4] Computer Science Department, University of Mines and Technology, Tarkwa, Ghana

Correspondence: Aderonke F. Thompson, Cyber Security Department, The Federal University of Technology, Akure, Nigeria.

Received: April 18, 2019 Accepted: January 7, 2020 Online Published: February 6, 2020

doi:10.5539/nct.v5n1p26 URL: https://doi.org/10.5539/nct.v5n1p26

Abstract

In the world of wireless communication, heterogeneous network topologies such as Wi-Fi and Long-Term Evolution (LTE) the topologies authentication service delivery forms a major challenge with access control; which is sought to be addressed. In this paper, we propose a security model by adapting Capability-based Context Aware Access Control (CCAAC) model for internet-enabled devices for defense against hacking or unauthorized access. The steps applied during the programming of this web application was followed through using the Elliptic-Curve Diffie–Hellman (ECCDH) algorithm so that the initiation of a random prime number between a range, the encryption and exchange of the devices public keys to the decryption are interpreted the right way to the machine making use of it. The results established a security model that has a good chance of being effective against present cyber-attacks other security loopholes.

Keywords: security, capability-based context aware access control model, elliptic-curve diffie-hellman algorithm

1. Introduction

As technology keeps on growing and new inventions of internet devices created, there is need to provide countermeasures for the protection of personal and business information, privacy and an assured integrity status of any internet-enabled host devices. A security system is a system that enforces boundaries between computer networks (Bishop, 2003). There has been a sporadic need of security for any and every computer-related system, more so internet-enabled devices. The very presence of crackers and hackers have made it mandatory for any organization producing or making use of computer systems to seek for the very best of cyber security. A successful attack launched on a computer system and its information content can compromise the confidentiality, integrity, and availability. Existing and emerging security challenges have posed a great threat to cyber information and access control. There are three major system security attacks in relation to the Internet of Things:

i. Denial of Service
ii. Man-In-The-Middle
iii. Replay attacker

DoS/DDoS (Denial of Service) cannot be totally prevented, but precautionary measures can be put in place so as to block unauthorized access or Phishing addresses from a host device standpoint (Mahalle *et al.,* 2013). Denial of Service is almost impossible to be eradicated and can better be worked upon if we were dealing with a host server. A man-in-the-middle attack occurs in a situation that there exists a user that sniffs information being sent in between the sender and receiver. A replay attack occurs, according to Microsoft (2017) when replays of the messages stream gotten by an attacker between two parties and to one or more of the parties is achieved. Otherwise mitigated, item redundancy is inevitable as the is seen as the legitimate messages from the computers.

An Internet-enabled device is a multimedia-capable mobile device that provides wireless Internet access. Examples of such devices are laptops, smartphones, tablets, amongst others. These devices provide entertainment, information and location-based services for users with real-time sharing and a 2-way communication features. Access control is required by all the devices and must be well protected in order to mitigate other viable threats in

the networks. Access grant for a predetermined time and its transaction is recorded. Otherwise, any attempt made also that suggest malicious entry is as well recorded, these could be useful during system auditing and log analysis to provide policies that will safeguard the systems against hackers, viruses, Distributed Denial of Service (DDOS) and so on. System should be protected to guard against attacks such as viruses that are more than 200 being discovered every month worldwide.

2. Related Works

2.1 Mutual Authentication Scheme in Secure Internet of Things Technology for Comfortable Lifestyle (Namje et al., 2015)

The authors proposed an inter-device authentication and session-key distribution system for devices with only encryption modules. In the proposed system, unlike existing sensor-network environments where the key distribution center distributes the key, each sensor node is involved with the generation of session keys. In addition, in the proposed scheme, the performance is improved so that the authenticated device can calculate the session key in advance. The proposed mutual authentication and session-key distribution system can withstand replay attacks, man-in-the-middle attacks, and wiretapped secret-key attacks. The system did not provide an encryption protocol, and used random nonce value of the hash message authentication code (HMAC), respectively, providing mutual authentication between lightweight devices with no HMAC and the data-transmission authentication for the transmitted data. Using a 256-bit key, which is longer than the result bit of the provided hash function, the system provides the same security level as HMAC. As the system generates a session key with a random nonce value, the entities can share a new session key for each authentication session. The responder can generate the session key in advance after transmitting the message (b), sending the transmission authentication message immediately for the data to be transmitted.

2.2 Connect-and-Protect: Building a Trust-Based Internet of Things for Business-Critical Applications (Aruba, 2015)

The author presented the combination of contextual data like location with rules and policies being essential for establishing trust in mobile IoT applications. The author described Aruba ClearPass Access Management System as being able to deliver consistent, scalable policy enforcement system-wide through sharing policies and threat notifications between Aruba's role-based firewall and other enforcement platforms. The architecture is based on the Aruba Adaptive Trust model, a defensive framework that leverages contextual Information from a multitude of sources to scrutinize user and device security posture before and after they connect. Adaptive Trust dispels the old notion of a fixed security perimeter that surrounds the physical network, which no longer applies because devices can now connect and exchange data from practically anywhere. Even devices with native standards-based Ethernet connectivity require a layered security approach because they can be exploited through cyber-attacks and malware. Connect-and-Protect security mechanisms Include; authenticating the source and destination devices; encrypting the packets for privacy; enveloping the packets inside a secure tunnel to ensure the integrity of the source, destination, and transport pathway; employing roles and context-based policies, modeled after the expected mode of operation, to trigger anomalous behavior alerts, quarantine wayward devices, and control parameters like network bandwidth and network services; inspecting northbound traffic with application firewalls and malware detection systems to monitor and manage behavior; and using mobile application management (MAM) and mobile device management (MOM) systems to monitor behavior and protect other devices in the event of a policy breach. Aruba controllers and controller-less access points incorporate a role-based firewall that manages network privileges and provides Identity-based auditing of activity. loT includes both stationary and mobile devices, as well as ones that switch between the two modes. This behavior means there is not always a fixed port through which a device always connects, so traditional firewalls that rely on port-based security are ineffective. Aruba's role-based firewall enforced controls based on user or device identity, not port, regardless of where or how the devices access the net-work. The role is applied during the authentication process, before the user or device has network access, using Active Directory. RADIUS, or LDAP comparable data. Unlike simple Access Control Lists (ACLs). Aruba's in-state firewall tracks upper-layer flows and ensures that unauthorized traffic can't bypass access control. For example, a packet claiming to be part of an established Telnet session would be blocked unless there was an actual established Telnet session. Firewall rules can be constructed based on identity, applications in use. source and destination of traffic, service type, time of day, physical location, and even device state when using client integrity software. Policy actions can include permit, deny, redirect to external devices or tunnels, logging, or quality-of-services actions such as setting Differentiated Services (DiffServ) bits and placing traffic into high- or low-priority queues. Automated blacklisting allows devices to be blacklisted from network access if firewall rules are violated even a single time Such a trip-wire is particularly useful for single-function sensors and actuators:

if the firewall detects a compromised device attempting to conduct unauthorized database queries or file server browsing, the device can be immediately disconnected from the device from the network and an alert generated.

2.3 IoT Privacy and Security Challenges for Smart Home Environments (Huichen et al., 2016)

The authors proposed a gateway architecture being the most appropriate for resource-constrained devices, and for high system availability. Two key technologies to assist system auto-management was identified. The first is the support for system auto-configuration which will enhance system security, and the second, the automatic update of system software and firmware is needed to maintain ongoing secure system operation. They also discussed three of the important and popular architectures, which are middleware, cloud and gateway architectures necessary for the security of the internet of Things.The gateway architecture comprises the ubiquitous sensor networks layer, the network layer and the service layer. IAGW includes a security module to implement the authentication, authorization and encryption. One of the benefits of this architecture is that it has a Quality of Service (QoS) module to prioritize traffic and guarantee resources for mission-critical operations. A systematic concept called Server-Based Internet-Of-Things Architecture (SBIOTA) is a proposed gateway server to provide an effective, efficient, secure and cooperative integration solution for IoT. This conceptual architecture includes a novel auto-configuration service on the gateway to facilitate the device's deployment and management process so that a device can be plugged into a network and be fully functional on that network with a minimum of manual configuration. Its initial approach is that the authentication and communication between the gateway and devices take place through a separate network port or a short-range antenna physically adjacent to the server. Before connecting the devices to the network, the user needs to place them in physical proximity to the gateway to be authenticated and exchange related information to ensure only legitimate devices are allowed to connect to the network .A gateway can implement sophisticated management algorithms on a reasonably powerful processor, and it can operate the critical Smart Home functions. Even in the temporary absence of an Internet connection, it can provide sophisticated firewall and proxy support to IoT devices so that they have minimal exposure to direct network attacks, and it can work with resource-constrained IoT devices without complex middleware. These are the two primary enhancements that are needed for a gateway-based Smart Home architecture to make it sufficiently secure for widespread adoption the authors suggested: auto-configuration support: It is expected that more and more smart household appliances will be interconnected to Smart Home networks. A lack of technical support is the biggest challenge in the household environment. Householders will be burdened by tedious, repetitive and error-prone manual tasks for adding and managing these smart devices on their home network, which can pose a major security risk. Therefore, for the successful implementation of a Smart Home, a secure auto-configuration approach should be further studied not only to simplify Smart Home device installation and maintenance but also to enhance the security in the auto-configuration process.

To achieve these supports, we propose an approach that requires functionality in the gateway and cloud-based services. When a new device is attached to the network, the gateway will use the device ID to interrogate a trusted web service to discover the details of the device—what its functionality is, what its commands are, what encryption and networking protocols it understands, and any essential firmware updates that are now available. This is a different approach to most auto-configuration approaches which require a lot of this information to be stored on the devices themselves, and for the devices to be able to already implement a deep protocol stack. With our approach, a simple device ID and a web service ensures this information is easily available and remains up-to-date.

3. System Framework

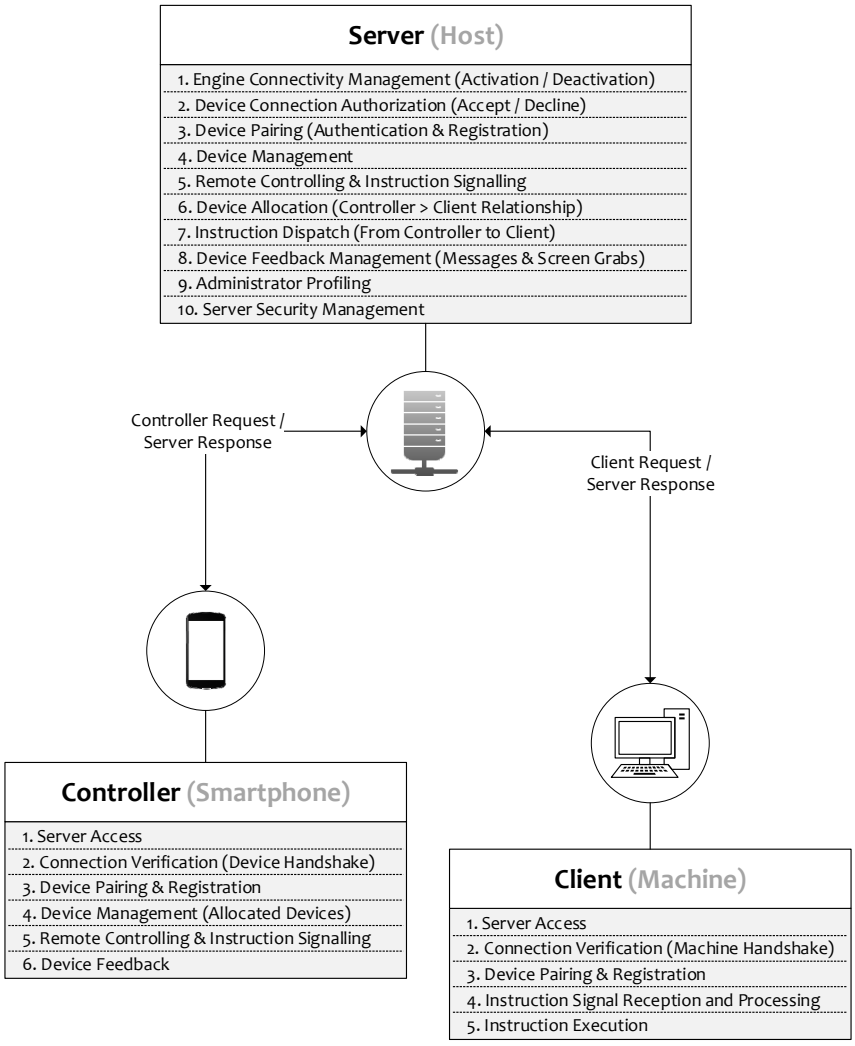

Figure 1. Architecture of the application system security

The server side deals with a number of activities including but not limited to engine connectivity management which encompasses the process of activation and deactivation, device connection authorization in terms of accepting or declining a client's or controller's request, and so on. The controller side, which are the smartphones, is the area where capability tokens implemented for the IACAC model takes effect and unique device identifier plus access rights are assigned to individual of the group of controllers that are already registered and present in the server side.

3.1 Verify Device

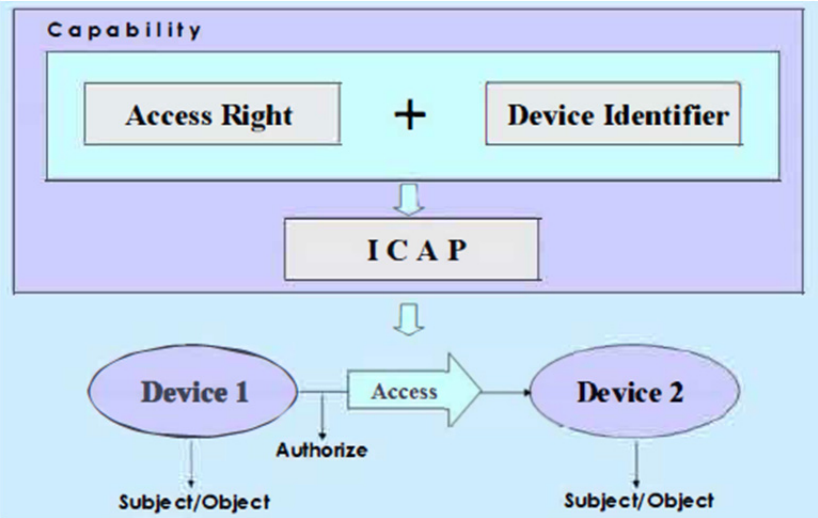

Figure 2. Capability Structure

At the backend, during verification of the device which is a controller (an admin), initial connection between server and the remote device is established, be it PC or Smartphone. The admin record is then fetched from the database, going through all of the available index. It checks if the server password typed in the provided text field correlates with what is in the database and subsequently gets the device if it already exists in the database. If the device properties are available or already exist as registered admin, only a log in operation is going to be allowed.

During signing up, it validates the username entered, if it passes then the server checks if the name provided is already taken or available. If taken, it prompts the user about it and if not, it goes on to check the access capability token.

3.2 Pairing Device

The complete CAC scheme is presented in Fig.6. It displays access based on CAC between two Wi-Fi devices. In this research, all devices are treated as subjects and resources to be accessed objects. The CAC implementation is considered as object for access with access rights (AR) defined as

$$AR = \{Read, Write, NULL\} \quad (3)$$

AR can take {Read}, {Write}, {Read, Write} or {NULL}. AR = {NULL}, implies permission denial. Verification against forgery is performed with a view to grant access to the device using the least privilege.

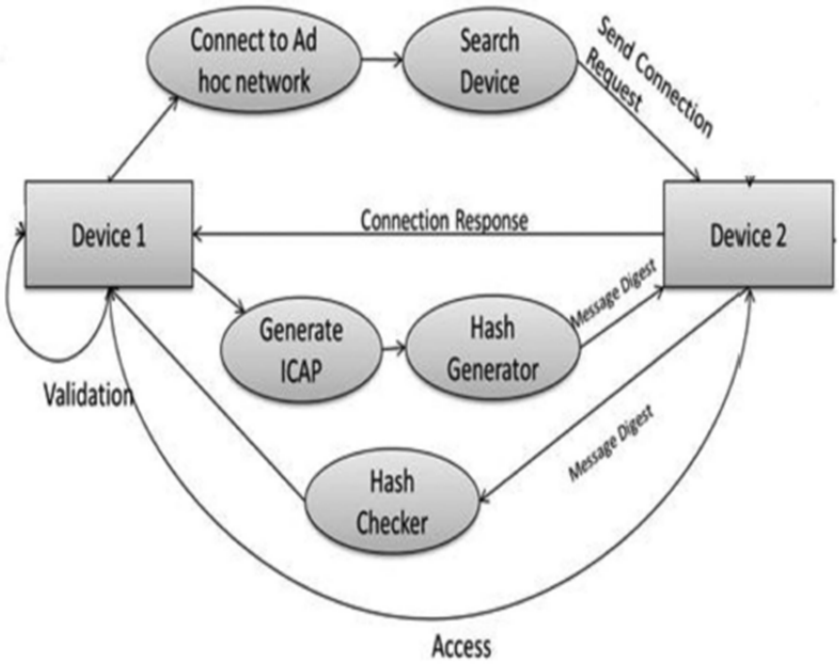

Figure 3. Device pairing access control functions of CAC (Capability based Access Control)

During device pairing at the backend, the pairing is called by Ajax to accept the server – device connection, ignore or delete it.

When the resulting number calculated by each device is calculated and displayed on either screens of the server and the controller or client after using their secret key to arrive at that particular number, the X_{uh} on both devices must be the same otherwise the super-admin can decline the attempted connection at his or her discretion. The buttons indicating “accept” or “decline” effectively carry out the instruction once pushed.

3.3 TetherRemote Application

The server in this research was simulated using a laptop for ease of implementation of the application tool used wherein lies the security model. The process begins with the server and a human super-admin. The super-admin is the official assigned to be the individual having all of the access controls, instructions and feedbacks the server is capable of. He/she can subsequently appoint admins and assign to them any possible combination of instructions they are allowed to perform using the controllers in their possession and this is after they would have registered those controllers to the servers before they are allowed to send instructions or see feedbacks from client machines.

But first, the TetherRemote web application (the implementation tool) is opened on a web browser as seen in Figure 4, using the current IP address of the server where all the menus and setting are available for the super-admin to work with. A preferred simulating server software Xampp assists with Apache and MySQL modules. These modules are started and the computer system is seen as the simulated server. This essentially serves as the primary node for networking of devices that are to be used.

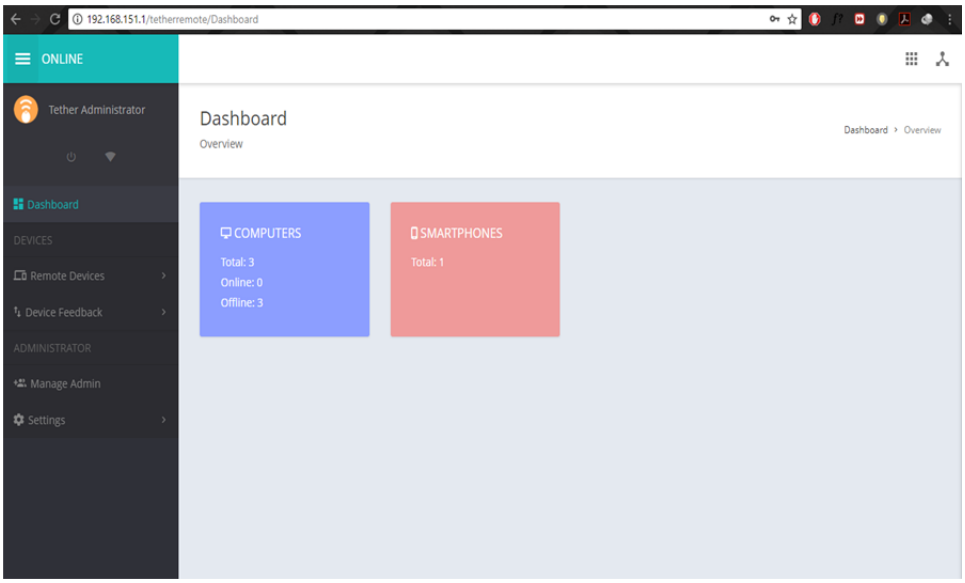

Figure 4. TetherRemote Web Application

The client-side TetherRemote application, which is plug-and-play, can be downloaded using a specific URL address and opened in a new client machine and a small window pops open revealing text fields where the server domain URL, the server password and the device identification is requested for such host device server connection. Meanwhile, the computer simulating the server has TetherRemote web application already opened, plus Xampp assisting with Apache and MySQL modules. The TetherRemote web application engine is then activated to allow connection and pairing from other internet-enabled host devices, be it smartphones or client computers.

3.4 Virtual Server

First, for this to work in our test environment, we will need some sort of virtual server that can stand-in as a replacement of a physical server, which therefore calls for the software called "Xampp". As shown in Figure 5, it contains a number of useful modules such as Apache, MySQL, Tomcat and so on for the efficient development of web applications without a physical server.

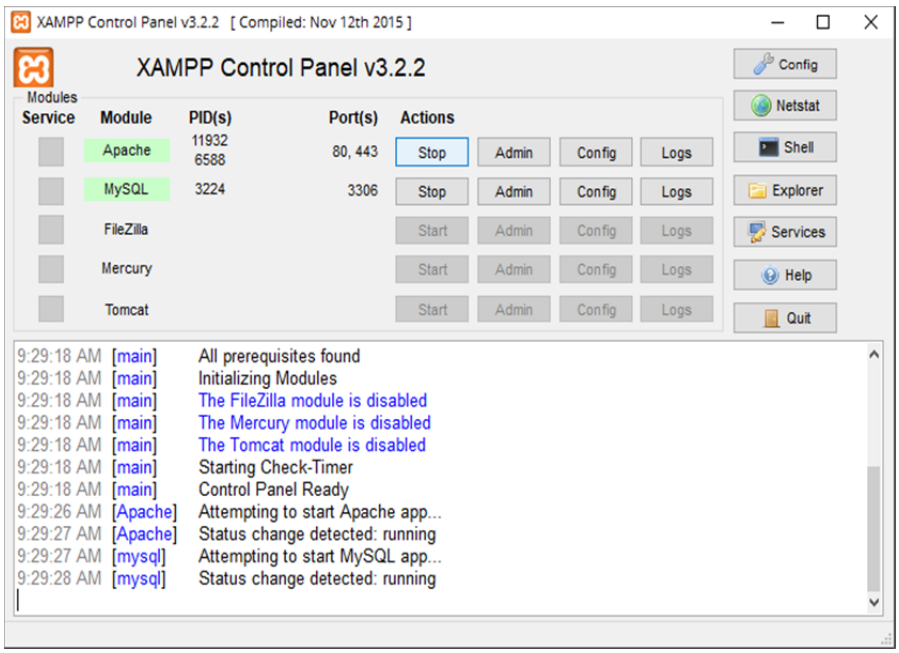

Figure 5. Server Simulation

The Xampp software is opened on the computer system that is to function as the server, and Apache plus MySQL modules are started. This helps TetherRemote to keep track of its database activities and handle connection requests from devices wanting to connect to, or pair with the server.

3.5 Pairing

The smartphone pairing and the client computer are the only two examples tested and during the initial step of pairing, either of the two will need to initiate the pairing procedure with a request to pair with the server using the server's current IP and password, and if it is the first time the smartphone is pairing with the server, there is a text field in the android TetherRemote application that gives you the avenue to do so. As for the client computer, the server automatically applies the name (computer name-PC) given to the computer as the display identity of that computer.

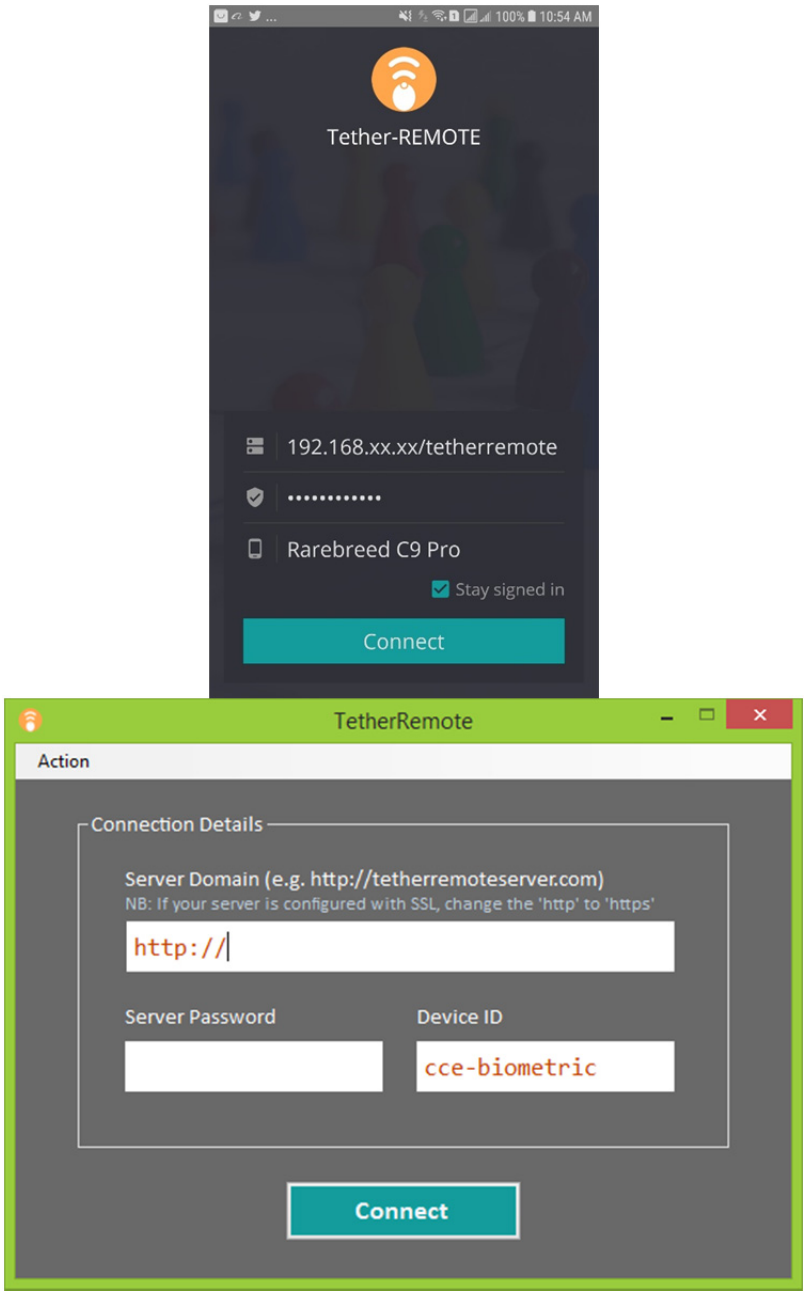

Figure 9. Smartphone controller and Client Machine log-in interface before pairing

On the server side during pairing, a corresponding pairing code pops up at the right side of the browser window web application to finalize the pairing procedure between the server and the controller or client computer.

3.6 Machine Assignment

After pairing, the client computers are listed in the Tether administrator smartphone controller menu where you can assign a number of client computers to a specific smartphone controller. These are all done on the TetherRemote web application page on a web browser. After a specific smartphone controller is selected for machine assignment, the checkboxes against each listed client machines can be ticked depending on the super-administrator's decision.

commands such as "View Machines" shows the currently assigned client machines to the smartphone controller and the "permissions" option brings up the interface where limitations can be set for each controller on the commands they are capable of sending to the client machines. Smartphone controllers can be as well deleted by the super-administrator.

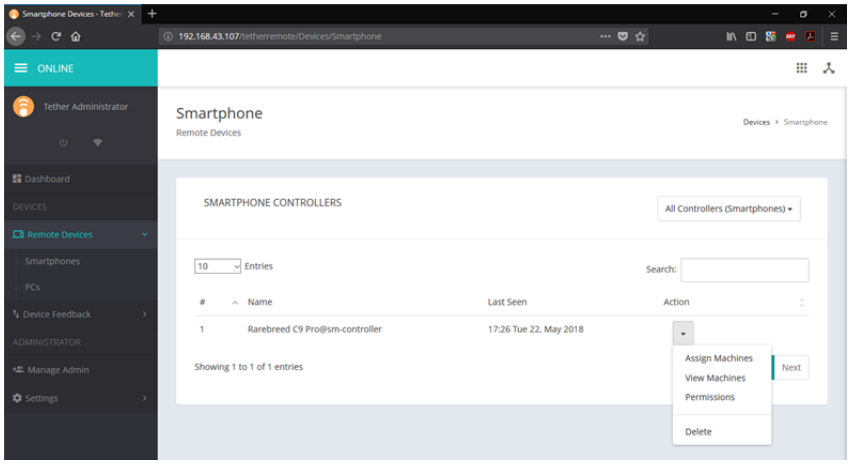

Figure 10. Smartphone Controller Interface on TetherRemote web application

4. Experiment Work

4.1 Access Control Solution Evaluation

In table 1, the context-aware Elliptic-curve Diffie–Hellman-based security authentication model proves to be an efficient way of validating the identity of devices trying to pair with a server and the table below illustrates how that is possible, and when compared with other available authentication solutions today, it stands out as the most effective way of ensuring there is enough trust between servers and machines to establish a connection.

Table 1. State-of-the-art evaluation summary

Solutions	Parameters						
			Attack Resistant				
	Mutual Authentication	Lightweight Solution	DoS	Man-in-the-Middle	Replay	Distributed Nature	Access Control
Context-aware ECCDH based Authentication	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Yes**	**Yes**
Authentication in Ad-hoc Wireless Network	No	No	Yes	Yes	Yes	No	No
Ubiquitous Access Control in MAGNET	No	No	No	No	No	Yes	Yes
ECC based Authentication in RFID	Yes	Yes	No	No	No	Yes	No
Progressive Authentication in Ad-hoc Networks	Yes	Yes	No	Yes	Yes	No	No
Authentication in loT	Yes	Yes	No	Yes	Yes	Yes	No
Peer Identification and Authentication	Yes	No	No	No	No	Yes	No

Table 2. Access control models Comparison

Models	Generic	Scalable	Granular	Delegation	Time Efficient	Security
ACL	Yes	No	No	No	No	No
RBAC	No	No	Yes	Yes	No	No
CWAC	Yes	No	Yes	No	No	No
CRBAC	Yes	No	Yes	Yes	No	No
CCAAC	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

In Table 2, the Capability-based Context Aware Access Control (CCAAC) is the most all-round access control having more than enough features to power applications involving authentication and good enough for the research to combat attacks from unauthorized entities.

5. Validation

Table 3. Computational Time for Identity and Capability based Context-Aware Access Control Scheme

Scheme	IACAC	HBQ	IoT_Auth
Authentication Time	$2D_H + 2D_{MAC} + 2D_{RC5}$	$2D_H + 2D_{MAC} + D_{RC5} + 3D_{MUL}$	$R + D_H + 2D_{MUL}$
Total	$2D_H + 2D_{MAC} + 2D_{RC5}$	$2D_H + 2D_{MAC} + D_{RC5} + 3D_{MUL}$	$R + D_H + 2D_{MUL}$
Total Time	14.02ms	2413.76ms	1604.07ms

Table 3 shows how quickly the response of the IACAC protocol goes through the mutual authentication process, which takes less time (14.02ms) than the other protocols ran on the Mica2 motes.

IoT Authentication scheme requires $R + D_H + 2D_{mul}$ time for mutual authentication which is approximately 1604.07 ms. The HBQ scheme takes $2D_H + 2D_{MAC} + D_{RC5} + 3D_{MUL}$ total time for authentication is 2,413.76 ms approximately (Chakravorty, 2006). It is worthy to note that both schemes do not address access control after authentication. IACAC takes only $D_H + 2D_{MAC} + 2D_{RC5}$ which takes only 14.02 ms which is enhanced than the other analyzed schemes. In IACAC, the $2D_H$ factor introduced comprises required time using one-way hash function.

6. Conclusions

The focus on this research was context and capability, which is a token. The presence of these features in a security model is essential enough for the experimentation and implementation of a programmed web application for

generic computer commands. Context in the forms of smartphone or computer identification demonstrates how the awareness of these two separate entities operate where one is a controller and the other is a client machine. Token on the other hand, is a rule that determines if a device is an admin, a super-admin or a client. All of these, when factored in, cumulates to a particular kind of security model with a context-aware capability-based access control having the proficiency to identify, authenticate and authorize devices with each having different parts to play. The steps applied during the programming of this web application was followed through using the Elliptic-curve Diffie–Hellman algorithm so that the initiation of a random prime number between a range, the encryption and exchange of the devices' public keys to the decryption are interpreted the right way to the machine making use of it.

With all these considered, it can be concluded that this researched security model has a good chance of being effective against cyber-attacks other security loopholes in our age.

References

Aruba Network Inc. (2015). *Connect-and-Protect: Building a Trust-Based Internet of Things for Business-Critical Applications*. White Paper 2015

Bishop, M. (2003). What is computer security? *IEEE Security & Privacy, 1*, Feb 2003. https://doi.org/10.1109/MSECP.2003.1176998

Chakravorty, R. A. (2006). Programmable Service Architecture for Mobile Medical Care. In *4th IEEE International Conference on Pervasive Computing and Communications*. 2006.

Huichen L., & Neil W. (2016). *IoT Privacy and Security Challenges for Smart Home Environments*. MDPI, Basel, Switzerland.

Mahalle, N. P., Anggorojati, B., Prasad, N. R., & Prasad, R. (2013). *Identity Authentication and Capability Based Acess Control (IACAC) for the Internet of Things*, February 2013. https://doi.org/10.1109/ANTS.2012.6524227

Microsoft docs, Replay Attacks (https://docs.microsoft.com/en-us/dotnet/framework/wcf/feature-details/replay-attacks), 30th March, 2017.

Namje, P., & Namhi, K. (2015). *Mutual Authentication Scheme in Secure Internet of Things Technology for Comfortable Lifestyle*（Department of Computer Education, Teachers College, Jeju National University, Jeju-si, Korea).

Copyrights