Updated SSDP Scheme for DDoS Attack Defense

Haiou Huang[12], Liang Hu[1] and Jianfeng Chu[1]
[1]College of Computer Science and Technology, Jilin University
Changchun, Jilin, China
[2]College of Electronic and Information Engineering., JiLin Agricultural Science and Technology University
Jilin, Jilin, China
[e-mail: hho_jl@163.com, chujf@jlu.edu.cn]
*Corresponding author: Jianfeng Chu

Received September 20, 2017; revised December 30, 2017; revised February 25, 2018;
accepted April 26, 2018; published September 30, 2018

Abstract

Abusing the Simple Server Discovery Protocol (SSDP) can induce an SSDP attack (including SSDP DoS, DDoS, DRDoS) posing a significant threat to UPnP devices. Rapid and extensive developments in computer technology, especially in regards to IoT, have made Upnp devices an indispensable part of our daily lives – but also render them susceptible to a variety of SSDP attacks without suitable countermeasures. This paper proposes the Two-dimensional table scheme, which provides high security at a reasonable computational cost. The feasibility and effectiveness of the proposed scheme are also validated by comparison against four other schemes (Stateless connections, Failing-together, Cookie, and Client puzzle).

Keywords: SSDP attack, Upnp devices, Two-dimensional Table

This work is funded by: National Key R&D Plan of China under Grant No. 2017YFA0604500, National Sci-Tech Support Plan of China under Grant No. 2014BAH02F00, and by Youth Science Foundation of Jilin Province of China under Grant No. 20160520011JH, and by Youth Sci-Tech Innovation Leader and Team Project of Jilin Province of China under GrantNo. 20170519017JH.

http://doi.org/10.3837/tiis.2018.09.026 **ISSN : 1976-7277**

1. Introduction

In the Black Hat Conference (2016), Elliott Peterson, co-founder of Malware Patrol, and Andre Correa, special agent at the FBI, provided notable insight on the current state of Internet-based Distributed Denial-of-Service (DDoS) attacks. According to their observations, UDP-based amplification attacks are particularly concerning. Surprisingly, some of the most highly-publicized DDoS attacks (e.g., DNS and NTP), which can generate hundreds of Gbps, were not the most used services – instead, SSDP was more common despite its low amplification factor. Peterson and Correa attribute this to the fact that most existing NTP and DNS servers have been patched in the recent past [1].

DDoS attacks are deployed in three separate phases. In the first stage, the attack is launched by forming a botnet of personal computers. What we call the second stage is a reflection attack through an open Internet server (such as NTP or DNS). The initiation of a reflection attack by exploiting the vulnerability of smart devices or IoT devices is the third phase.

Users may become quickly aware of an attack in the first or second stage, as virus protection software, other patches, and repairs to holes in open Internet servers (such as NTP or DNS) can effectively prevent them. Recent hackers have found new ways to attack other devices, however, such as printers, home routers, web cameras, and smart appliances. These devices follow UPnP protocols for network communication: they discover other UPnP devices, leverage the SSDP of source port 1900, and interact with each other accordingly. For this reason, the proposed scheme was designed with special consideration to the third phase of DDoS attacks.

SSDP reflection attacks, a relatively new category of DDoS attack, have grown increasingly problematic. Akamai’s 2015 Q1 State of the Internet/Security Report [2] shows that the SSDP Reflection Attack is currently the most prevalent DDoS (20.78%). SSDP attacks only accounted for 14% of the total in the fourth quarter of 2014. This striking change is illustrated in **Fig. 1**. And as shown in **Fig. 2**, the SSDP protocol retained its top spot as the single largest number of reflector source in Q2 [3].

Similar to NTP and DNS reflection attack, the principle of a SSDP reflection attack is that the attacker forges the victim’s IP, then client issues SSDP requests to a massive amount of smart devices. Smart devices which receive these requests send response packets to the victim based on the source IP. AliCloud was attacked in this manner in June of 2015, during which time 80% of the attack was in the form of an SSDP reflection attack [4].

As shown later, the amplification of SSDP attack is 30.8 [5], i.e., is much smaller than the amplification of NTP, Chargen, or others similar. Of course, there is a huge quantity of smart devices on the Internet, however; the rapid and continual increase in this number of online smart devices alongside advancements in IoT creates greater susceptibility to such attacks.

1) The quantity of printers, home routers, and other intelligent home appliances is, as discussed above, increasing daily and across the globe. Unlike NTP Servers which are managed by professionals, these devices are managed by individual users and can thus be difficult to protect.

2) The costs of an SSDP reflection attack are much lower than traditional DoS attack methods, so they are relatively easy to launch.

3) The vulnerability of SSDP protocol is the essential cause of an SSDP reflection attack.

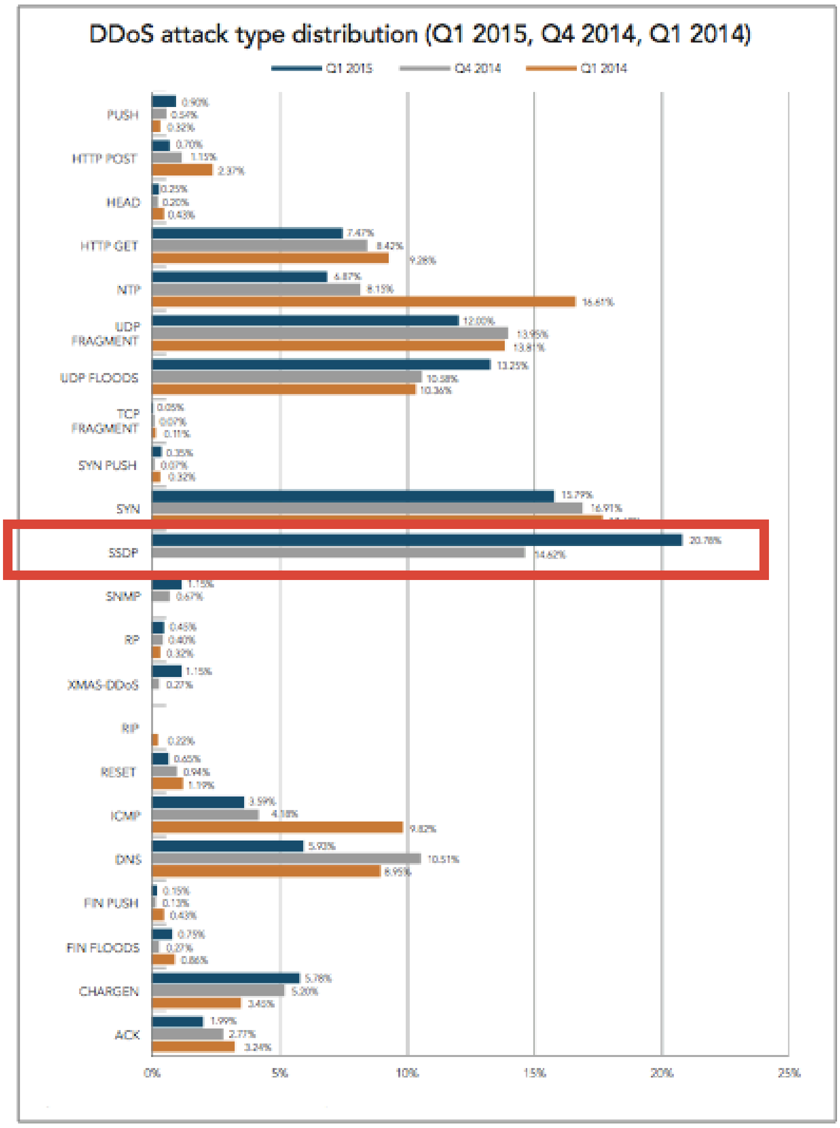


Fig. 1. DDoS attack vector frequency observed over the Akamai's PLX routed network [2]

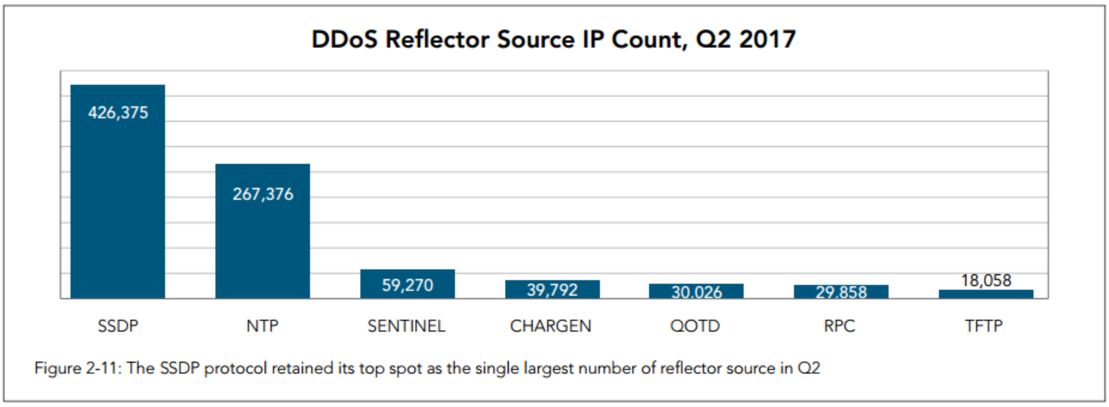


Fig. 2. DDoS Reflector Source IP Count, Q2 2017[3]

The primary causes of SSDP DDoS attacks can be summarized as follows.

Some would assert that preventing IP address spoofing is sufficient to prevent Dos attacks [6] [7], but it is simply impossible to ensure that all UPnP device users install BCP38 (Best Current Practice). Most SSDP servers are not under the control of professionals. Secure Service Configuration, conversely, has a negative impact on SSDP servers (though shutting off NTP servers will not influence users' professional or personal lives.)

This paper discusses the essential cause of SSDP attacks (including DoS, DDoS and DRDoS) with special focus on why exactly it is so difficult to block them. We then compare various existing methods to mitigate DoS attacks and highlight their various defects before introducing the proposed scheme. We compare the proposed method against four other state-of-the-art methods in regards to communication overhead, security, computation costs of the server, computation costs to the client, and server storage costs. We also serialize the response packets from SSDP servers, as discussed in detail below.

Our contributions can be summarized as follows.

1) We analyze common methods of mitigating DoS attacks in detail to point out several defects in them.

2) We introduce the Two-dimensional table scheme.

3) We compare the proposed scheme against four others similar in regards to communication overhead, security, computation costs of the server, computation costs of the client, client storage costs, and server storage costs to validate the feasibility and effectiveness of Two-dimensional table.

4) We serialize the response packets from SSDP servers.

The remainder of this paper is organized as follows. Section 2 discusses the extant research on this subject; Section 3 provides definitions and background knowledge relevant to the topic. The proposed scheme, Two-dimensional table, is described in detail in Section 4. Section 5 reports our comparative analysis of the proposed scheme and four others in terms of security and performance. Section 6 discusses our conclusions.

2. Background

This section discusses the differences among DoS, DDoS, and DRDoS followed by a description of SSDP and analysis of the SSDP attack principle.

2.1 DoS, DDoS, and DRDoS

DoS (Denial of Service), DDoS (Distributed Denial of Service), and DRDoS (Distributed Reflection Denial of Service), a transformation of DDoS, differ only in that DRDoS does not need to occupy any particularly large amount of chickens (also known as Puppet Machines, are machines that can be remotely controlled by hackers.). If the victim has begun SSDP service, the attacker can move on the victim directly – that is, launch a DoS or DDoS – otherwise, the attacker sends a mass of packets carrying the victim's IP address to attack a host [8] [9], then the attack host generates a large amount of responses to the source IP address (victim) leading to denial of service as a DRDoS. The principles of these three attacks are shown in **Figs. 3-5**.

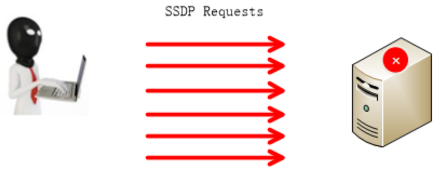


Fig. 3. DoS principle

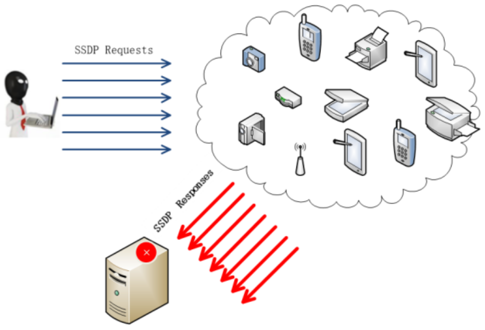


Fig. 5.

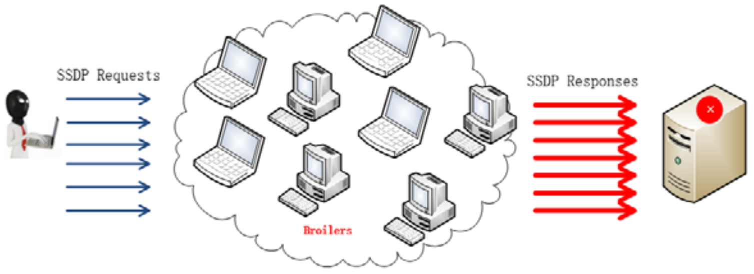


Fig. 4. DDoS principle

2.2 What is SSDP?

The Simple Service Discovery Protocol (SSDP) is an application-layer protocol and one of the core protocols of UPnP [10]. The protocol offers the user a mechanism for managing and maintaining network services without any necessary configuration. It works via multicast discovery method based on notifying and discovering routes. The client discovers service at the multicast address 239.255.255.250:1900 [11], while SSDP over IPv6 uses the address ff0X::C [12]. All the devices that offer service monitoring perform service discovery requests simultaneously. If a server monitors any request that matches its service, it will respond in a unicast manner.

Common protocol requests can be divided into two categories: 1) service notification (NOTIFY method), which can be used by devices (SSDP servers) to announce their presence information, and 2) search request (M-SEARCH method), through which the client can discover available services on a given network. The request information usually contains special device or service data such as the device type, identifier, and/or URL address of the point-to-device description document.

2.3 SSDP Attack Principle

SSDP permits networked devices like Wi-Fi access points, Internet gateways, IP cameras, personal computers, and smart mobile devices to discover each other's presence on the network and establish functional network services for data sharing, communication, and entertainment [13].

The Simple Object Access Protocol (SOAP) provides a standard under which different applications run on different operating systems, and uses various technologies and programming languages which can communicate with each other. It is used to deliver control messages to UPnP devices and to pass information back from the devices. Attackers have discovered that SOAP requests can be crafted to elicit a response that reflects and amplifies a packet, which can then be redirected towards a target. By employing a large number of devices, attackers create large quantities of attack traffic that can be aimed at their selected targets [9][13]. When the client initiates a request, the response packets are usually much greater in abundance than the request packets.

As mentioned above, the bandwidth amplifier factor of SSDP attack is 30.8. To explain how we arrived at this value, it is first necessary to define BAF and PAF.

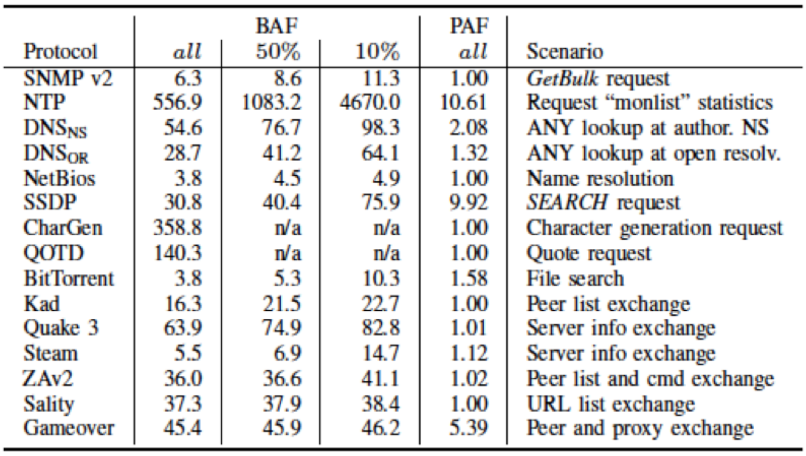
The bandwidth amplification factor (BAF) is a bandwidth multiplier defined by the UDP payload bytes that an amplifier sends to answer a request compared to the number of UDP payload bytes of the request [5].

$$BAF = \frac{len(UDP\ payload)\ amplifier\ to\ victim}{len(UDP\ payload)\ attacker\ to\ amplifier} \tag{1}$$ [5]

The packet amplification factor (PAF) is the packet multiplier in terms of the number of IP packets that an amplifier sends to answer a request.

$$PAF = \frac{number\ of\ packets\ amplifier\ to\ victim}{number\ of\ packets\ attacker\ to\ amplifier} \qquad (2)\ [5]$$

The BAF of SSDP is shown in **Fig. 6**; SSDP's BAF is 30.8 and its PAF is 9.92.

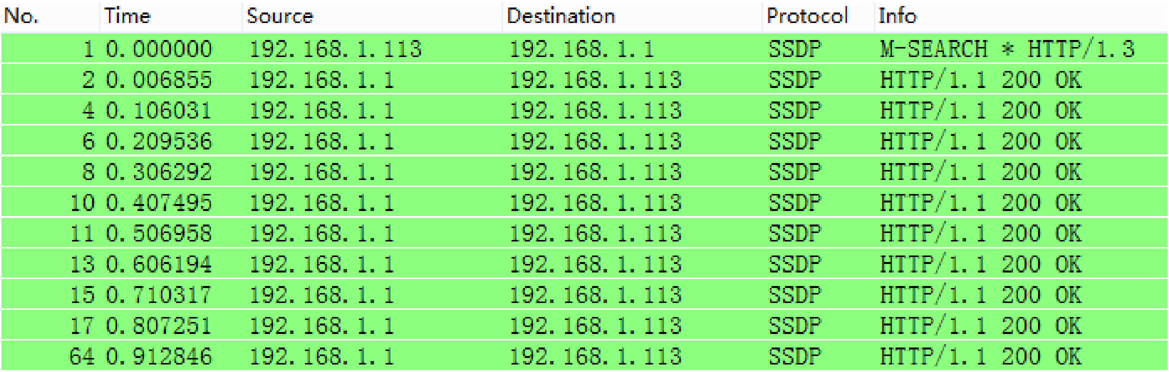


	BAF			PAF	
Protocol	*all*	50%	10%	*all*	Scenario
SNMP v2	6.3	8.6	11.3	1.00	*GetBulk* request
NTP	556.9	1083.2	4670.0	10.61	Request "monlist" statistics
DNS_{NS}	54.6	76.7	98.3	2.08	ANY lookup at author. NS
DNS_{OR}	28.7	41.2	64.1	1.32	ANY lookup at open resolv.
NetBios	3.8	4.5	4.9	1.00	Name resolution
SSDP	30.8	40.4	75.9	9.92	*SEARCH* request
CharGen	358.8	n/a	n/a	1.00	Character generation request
QOTD	140.3	n/a	n/a	1.00	Quote request
BitTorrent	3.8	5.3	10.3	1.58	File search
Kad	16.3	21.5	22.7	1.00	Peer list exchange
Quake 3	63.9	74.9	82.8	1.01	Server info exchange
Steam	5.5	6.9	14.7	1.12	Server info exchange
ZAv2	36.0	36.6	41.1	1.02	Peer list and cmd exchange
Sality	37.3	37.9	38.4	1.00	URL list exchange
Gameover	45.4	45.9	46.2	5.39	Peer and proxy exchange

Fig. 6. BAFs per protocols [5]

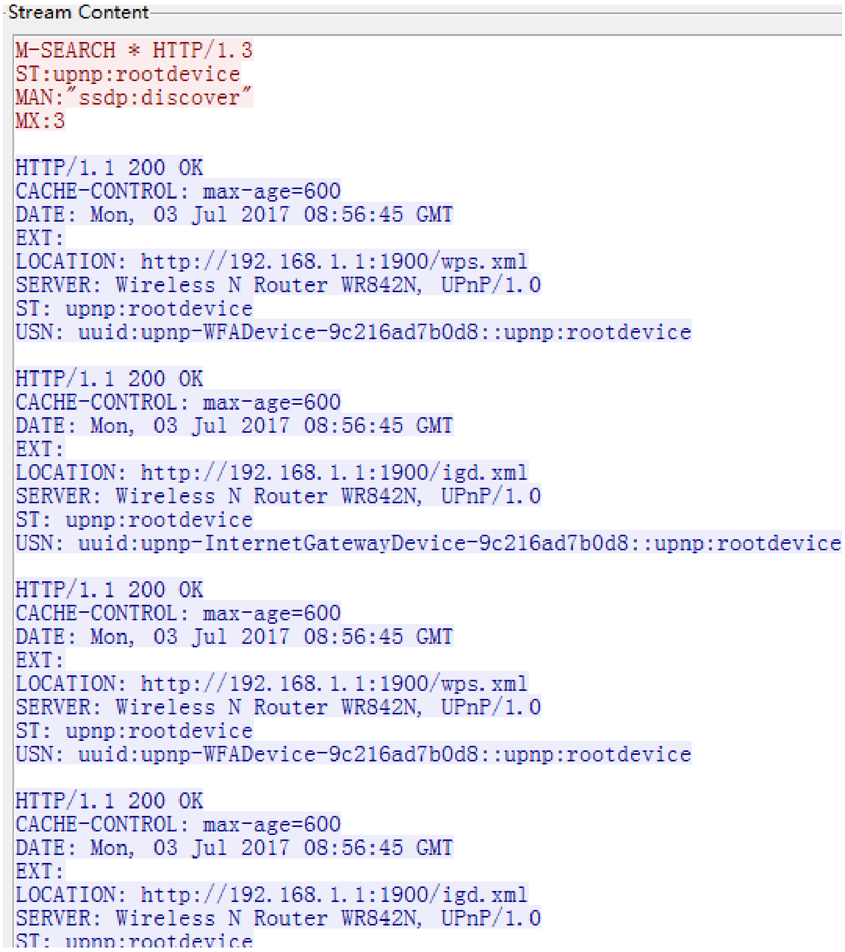
"All" marks the average BAF of all amplifiers, and "50%" and "10%" mark the average BAF when using the worst 50% or 10% of the amplifiers, respectively.

To support our overview of the single SSDP reflection attack shown below (**Figs. 7-10**), one requested packet of 112 bytes received 10 response packets totaling 3080 bytes. This is 10x in packet count, or astoundingly, about 30x in byte count – in the laboratory, on average, we could obtain about 10x in packet count, and each response packet is about 3x in byte count, making a total of 30x in byte count.



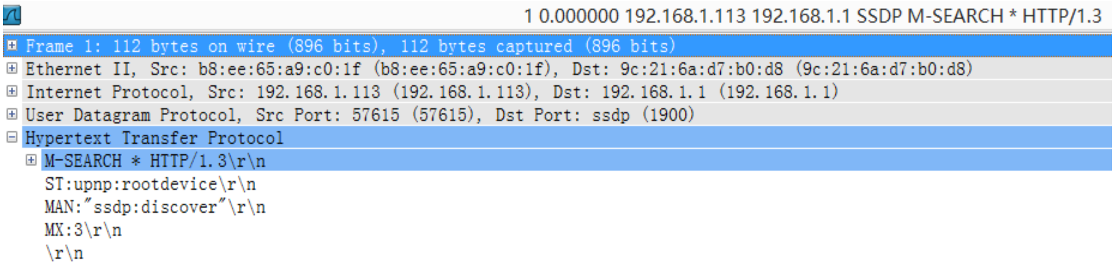
No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.113	192.168.1.1	SSDP	M-SEARCH * HTTP/1.3
2	0.006855	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
4	0.106031	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
6	0.209536	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
8	0.306292	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
10	0.407495	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
11	0.506958	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
13	0.606194	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
15	0.710317	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
17	0.807251	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
64	0.912846	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK

Fig. 7. One search request responded to by 10 packets



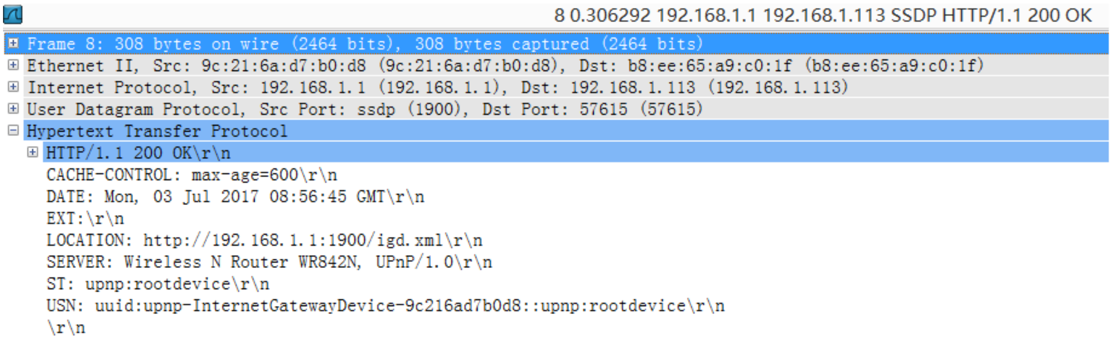
```
M-SEARCH * HTTP/1.3
ST:upnp:rootdevice
MAN:"ssdp:discover"
MX:3

HTTP/1.1 200 OK
CACHE-CONTROL: max-age=600
DATE: Mon, 03 Jul 2017 08:56:45 GMT
EXT:
LOCATION: http://192.168.1.1:1900/wps.xml
SERVER: Wireless N Router WR842N, UPnP/1.0
ST: upnp:rootdevice
USN: uuid:upnp-WFADevice-9c216ad7b0d8::upnp:rootdevice

HTTP/1.1 200 OK
CACHE-CONTROL: max-age=600
DATE: Mon, 03 Jul 2017 08:56:45 GMT
EXT:
LOCATION: http://192.168.1.1:1900/igd.xml
SERVER: Wireless N Router WR842N, UPnP/1.0
ST: upnp:rootdevice
USN: uuid:upnp-InternetGatewayDevice-9c216ad7b0d8::upnp:rootdevice

HTTP/1.1 200 OK
CACHE-CONTROL: max-age=600
DATE: Mon, 03 Jul 2017 08:56:45 GMT
EXT:
LOCATION: http://192.168.1.1:1900/wps.xml
SERVER: Wireless N Router WR842N, UPnP/1.0
ST: upnp:rootdevice
USN: uuid:upnp-WFADevice-9c216ad7b0d8::upnp:rootdevice

HTTP/1.1 200 OK
CACHE-CONTROL: max-age=600
DATE: Mon, 03 Jul 2017 08:56:45 GMT
EXT:
LOCATION: http://192.168.1.1:1900/igd.xml
SERVER: Wireless N Router WR842N, UPnP/1.0
ST: upnp:rootdevice
```

Fig. 8. M-search request (red) and reply (blue)



```
1 0.000000 192.168.1.113 192.168.1.1 SSDP M-SEARCH * HTTP/1.3
⊞ Frame 1: 112 bytes on wire (896 bits), 112 bytes captured (896 bits)
⊞ Ethernet II, Src: b8:ee:65:a9:c0:1f (b8:ee:65:a9:c0:1f), Dst: 9c:21:6a:d7:b0:d8 (9c:21:6a:d7:b0:d8)
⊞ Internet Protocol, Src: 192.168.1.113 (192.168.1.113), Dst: 192.168.1.1 (192.168.1.1)
⊞ User Datagram Protocol, Src Port: 57615 (57615), Dst Port: ssdp (1900)
⊟ Hypertext Transfer Protocol
  ⊞ M-SEARCH * HTTP/1.3\r\n
    ST:upnp:rootdevice\r\n
    MAN:"ssdp:discover"\r\n
    MX:3\r\n
    \r\n
```

Fig. 9. An M-search request in which 192.168.1.113 sends to 192.168.1.1



```
8 0.306292 192.168.1.1 192.168.1.113 SSDP HTTP/1.1 200 OK
⊞ Frame 8: 308 bytes on wire (2464 bits), 308 bytes captured (2464 bits)
⊞ Ethernet II, Src: 9c:21:6a:d7:b0:d8 (9c:21:6a:d7:b0:d8), Dst: b8:ee:65:a9:c0:1f (b8:ee:65:a9:c0:1f)
⊞ Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.113 (192.168.1.113)
⊞ User Datagram Protocol, Src Port: ssdp (1900), Dst Port: 57615 (57615)
⊟ Hypertext Transfer Protocol
  ⊞ HTTP/1.1 200 OK\r\n
    CACHE-CONTROL: max-age=600\r\n
    DATE: Mon, 03 Jul 2017 08:56:45 GMT\r\n
    EXT:\r\n
    LOCATION: http://192.168.1.1:1900/igd.xml\r\n
    SERVER: Wireless N Router WR842N, UPnP/1.0\r\n
    ST: upnp:rootdevice\r\n
    USN: uuid:upnp-InternetGatewayDevice-9c216ad7b0d8::upnp:rootdevice\r\n
    \r\n
```

Fig. 10. One of the replies from 192.168.1.1 to 192.168.1.113

3. Related Work

Those attacks called DoS, DDoS, and DRDoS all leverage the vulnerability of protocols, so the solutions to these attacks are fairly similar. There are three main countermeasures to DoS (including DDoS, DRDoS) attacks which currently exist: Stateless connections, Increasing expense, and Weak authentication. None of them may be directed against SSDP DDoS, however.

Aura developed Stateless connections [14] to relieve the memory burden of a responder. Upon receiving an initiator's request, the responder performs this protocol without saving protocol-relative state information and takes state information as part of the response message; the initiator returns the state information to the responder. In other words, the responder saves the state message at initiator before ensuring the initiator's identity.

Failing-together was proposed by Matsuura and Imai in *Protection of Authenticated Key-agreement Protocol against a Denial-of-Service Attack* [15][16]. It can be applied to update IKE (Internet key exchange) to actively defend against DoS [17][18][19]. The principle of Failing-together is to try to reduce the responder's computation load while raising that of the initiator. When the computing power of both sides is comparable, computation of the initiator is greater than or equal to the computation of the responder. This method is mainly applicable to special signatures and signature verification algorithms [20][21] [22][23].

Dwork and Naor proposed Proof of Work [24], which later inspired Jules to create Client puzzle [25] for defense against DoS attacks. The responder does not save any state upon receiving a request, but sends a cryptography problem; if the initiator answers correctly, the remainder of the protocol is then implemented.

Photuris [26] was put forward by Karn and Simpson, and Cookie has been applied in a number of security protocols since. Essentially, Cookie is a weak authentication of a stateless connection. The initiator first issues a request, then obtains one cookie which can only be made or verified by the responder; it then initiates the connection with the responder again and sends another cookie which is examined by the responder. If it is valid, the responder trusts that the initiator is not an attacker and offers service as protocol.

A stateless connection preserves memory resources, but at the cost of increased computation cost. It also violates the encryption principle under which secure cryptographic protocols are designed [27]. Failing-together ensures that the initiator's and responder's computation loads are comparable, but can only block DoS attacks – it is ineffective for DDoS. It also necessitates high memory and computation overhead, and cannot block replay attacks. Replay attacks also occur easily when using the Client puzzle method. Further, it is difficult to set sufficiently challenging cryptographic problems. The Cookie method requires generating a new cookie every communication, so the communication overhead is excessive.

In short: there is urgent demand for new defenses against SSDP attacks.

4. Proposed Scheme

We propose defining a two-dimensional table to store server and client information including the time at which the client initiates the request and the number of successive requests from client to server, so we call the proposed scheme "Two-dimensional Table".

4.1 Two-dimensional Table

The proposed scheme works in the following step-wise process.

Step 1 (Initialization): Assume that there is no client sending any requests to the server. The server generates a random number named SN and saves it locally, then defines a variable length array C[X] which counts the number of requests from client to server as well as an empty two-dimensional space to save the request time RTi (i.e., the unique identification of every client), C[i] (the number of successive requests from client i to the server); hash value (HASHi) is created by the server.

Step 2: Client i initiates a request to the server.

Step 3: The server judges whether RTi is new; if so, assume that the CIPi is the IP address of the client, then C[i] is set to 1 to generate the hash value HASH(RTi, SN, CIPi), and RTi and C[i] and hash values are summed as a new record entry to the two-dimensional table. RTi and HASHi are then sent to the client (otherwise step 5 begins).

Step 4: Client i sends Requesti || HASHi || RTi to the server.

Step 5: The server seeks RTi and HASHi values from the two-dimensional table that match client i and compares them against authentication values sent by client i.

If validation fails, then the server marks the client and refuses any further requests from it; Otherwise, the server judges whether C[i] is greater than N (the maximum number of successive requests from client to server). If C[i] <=N, the server satisfies the needs of client i and continues the protocol; Otherwise, the server clears the record that matches RTi, then goes to step 3.

(Note1: If a client initiates a request to the server but provides the wrong validation messages, then the server marks the client and refuses any further requests from it. This can prevent adversaries from launching resource exhaustion attacks by leveraging SSDP's retransmission timeout mechanism. Note: The response packets far exceed the request packets, as response data is three times as large as request data. We propose that the server serialize the response packet before sending it to the client, which can control response packet amplification and mitigate SSDP attacks.)

Algorithm flow diagram of Two-dimensional Table is shown below(**Fig. 11**)：

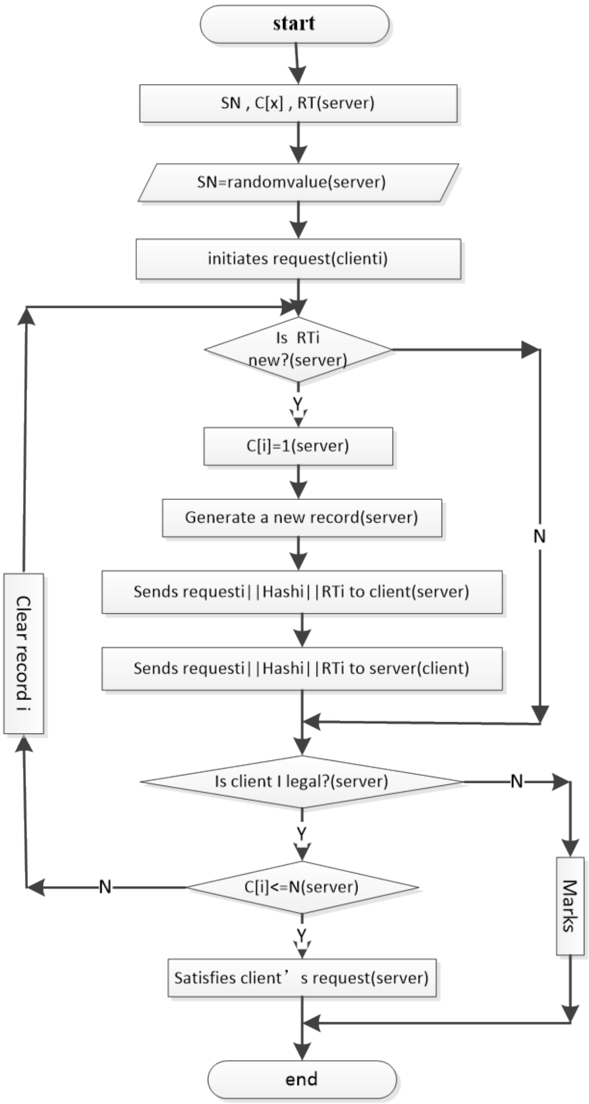

start
SN , C[x] , RT(server)
SN=randomvalue(server)
initiates request(clienti)
Is RTi new?(server)
Y
N
C[i]=1(server)
Generate a new record(server)
Sends requesti||Hashi||RTi to client(server)
Sends requesti||Hashi||RTi to server(client)
Clear record i
Is client I legal?(server)
N
Y
Marks
C[i]<=N(server)
N
Y
Satisfies client' s request(server)
end

Fig. 11. Algorithm flow diagram of Two-dimensional Table

Fig. 12 shows the process through which client i sends a request to the server the first time and establishes a connection to the server successfully. After this has been achieved, if client i has a request, client i needs only to send HASH (RTi, SN, CIP**i**) and the request together to the server providing that C[i] is not greater than N. This does not introduce any additional correspondence time.

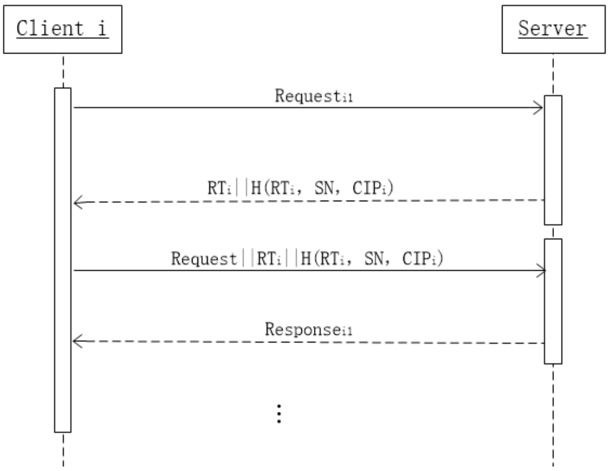


Fig. 12. Connection established for the first time

4.2 Advantages of Two-dimensional Table

The advantages of Two-dimensional table can be summarized as follows.

1) It saves not only memory, but also computation. This is shown in row 3, row 4, row 5 and row6 in **Table 1**.
2) Achieving verification via one-way hash function conforms to principles for secure cryptographic protocol design [21].
3) The server can identify different clients through request time RT_i and distinguish clients in the local net effectively.
4) There is one-to-one correspondence between RT_i and hash value $HASH_i$. Because if the number of successive requests that client i sends to server C[i] is greater than N, the record i will be cleared and a new record appended, so the record i is fresh – this would block any replay attack.
5) For the same client, within N time requests, the server needs twice as much correspondence than the original protocol to establish a connection with client i and to verify the client's identity upon only the initial contact. Compared to the other schemes discussed above, this gives ours relatively low communication overhead and guarantees lightweight and negotiable UDP.
6) The proposed method is also robust to resource exhaustion attacks. If a client does not pass the server's validation, it cannot originate a DoS but can make use of overtime and

error retransmission to initiate a resource exhaustion attack. We propose that if one client gives the wrong verification information, it is marked and its further requests rejected to prevent such attacks.

7) The server serializes the response packet before sending it to the client, which can mitigate SSDP attacks.

4.3 Security analysis of current schemes and proposed scheme

As mentioned above, Stateless Connection, Failing-together, Client Puzzle, and Cookie are the four main schemes currently utilized to resolve DoS/DDoS/DRDoS attacks. The security of each scheme is discussed below.

According to the literature [14] and the description above (Related Work section), it is not difficult to analyze the safety hazards of Stateless Connection: the attacker can replay Msg3 and launch a DoS attack. Cookie, while preventing most IP spoofing attackers, is vulnerable to three other types of attack [20]. First, an attacker may use a real IP address. Second, an attacker may use a forged IP address to issue a request and obtain cookies by tapping on the returned path, using IP source routing, and performing a DoS attack with the cookie and a false IP address. Third, an attacker may eavesdrop on a cookie sent to an IP, then use the cookie to forge the IP for a DoS attack. For Client Puzzle, attackers can replay the Nc and solution [19] to initiate a DoS attacks. Failing-together uses the signature and signature verification method [15] [16], which can effectively prevent DoS attacks but is yet vulnerable to DDoS attacks.

Two-dimensional Table involves a different approach to resource depletion and replay attacks. Because the number of a client uses the same verification code to control visits to the server within N times, if exceeded, The server will regenerate a new record for the client and assign a new verification code when the client accesses the server N+1 times. So in a short time, the client won't have many requests. What’s more, if the client provides the wrong authentication code, the service side immediately marks the client as an illegal intruder and terminates the response to any of its requests. Thus, the client can't cause resource depletion and replay attacks on the server. So, for the SSDP reflection attack, compare to other schemes, Two-dimensional Table is safer.

4.4 Security verification of the proposed scheme

In order to test the security of the proposed scheme, we do an experiment.

4.4.1 Experimental settings

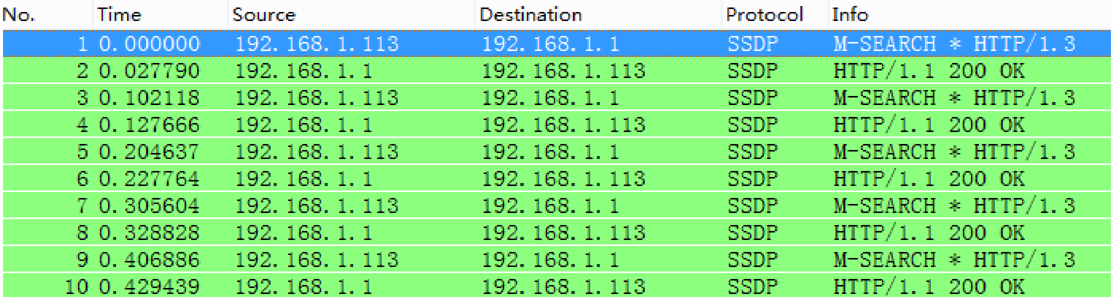
We conducted a simple experiment to verify the security of Two-dimensional Table. The equipment used in the experiment was a normal home router (model TL-WR841N) and computer running the Windows 8 operating system.

4.4.2 Results and analysis

Consider the SSDP reflection attack process. It is assumed that the home router IP address is 192.168.1.1 and the request to the router uses the forgery IP192.168.1.113. We found that one request (from 192.168.1.113) acquires a response of 10 packets from the router. The specific experimental results are shown in **Fig. 6**. As we described in the Background section, the response is 30 times that of the request in terms of byte count. This result and the description from the literature [5], "the average bandwidth amplifier factor of SSDP attack is 30.8", are consistent.

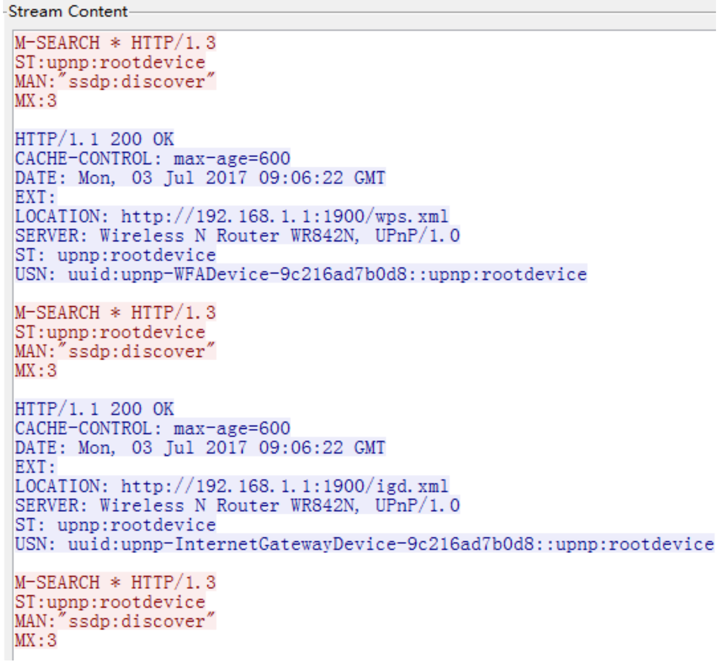
We conducted a follow-up verification experiment on Two-dimensional Table according to the algorithm flow discussed in Section 4.1.

Experimental results are provided in **Figs. 13-14**. When 192.168.1.1 receives one packet from 192.168.1.113, it only sends one packet as a reply. This suggests that the proposed scheme is indeed an effective countermeasure against SSDP reflection attacks.



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.113	192.168.1.1	SSDP	M-SEARCH * HTTP/1.3
2	0.027790	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
3	0.102118	192.168.1.113	192.168.1.1	SSDP	M-SEARCH * HTTP/1.3
4	0.127666	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
5	0.204637	192.168.1.113	192.168.1.1	SSDP	M-SEARCH * HTTP/1.3
6	0.227764	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
7	0.305604	192.168.1.113	192.168.1.1	SSDP	M-SEARCH * HTTP/1.3
8	0.328828	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK
9	0.406886	192.168.1.113	192.168.1.1	SSDP	M-SEARCH * HTTP/1.3
10	0.429439	192.168.1.1	192.168.1.113	SSDP	HTTP/1.1 200 OK

Fig. 13. One-search request responded by one packet



Stream Content

```
M-SEARCH * HTTP/1.3
ST:upnp:rootdevice
MAN:"ssdp:discover"
MX:3

HTTP/1.1 200 OK
CACHE-CONTROL: max-age=600
DATE: Mon, 03 Jul 2017 09:06:22 GMT
EXT:
LOCATION: http://192.168.1.1:1900/wps.xml
SERVER: Wireless N Router WR842N, UPnP/1.0
ST: upnp:rootdevice
USN: uuid:upnp-WFADevice-9c216ad7b0d8::upnp:rootdevice

M-SEARCH * HTTP/1.3
ST:upnp:rootdevice
MAN:"ssdp:discover"
MX:3

HTTP/1.1 200 OK
CACHE-CONTROL: max-age=600
DATE: Mon, 03 Jul 2017 09:06:22 GMT
EXT:
LOCATION: http://192.168.1.1:1900/igd.xml
SERVER: Wireless N Router WR842N, UPnP/1.0
ST: upnp:rootdevice
USN: uuid:upnp-InternetGatewayDevice-9c216ad7b0d8::upnp:rootdevice

M-SEARCH * HTTP/1.3
ST:upnp:rootdevice
MAN:"ssdp:discover"
MX:3
```

Fig. 14. M-search request (red) and reply (blue)

5. Performance Analysis

The section above describes the Two-dimensional Table scheme in detail as well as our proof that it is more secure than other existing schemes against SSDP reflection attacks. However, this is not sufficient to illustrate the advantages of the Two-dimensional Table scheme.

In order to further verify the feasibility of Two-dimensional table, we compared our scheme against the four schemes we mentioned in Related Works in regards to five indexes (communication overhead, server computation costs, client computation costs, server storage costs, and client storage costs). All the data in the table below (**Table 1**) are additional data based on SSDP protocol with N communications between the server and client.

The second row of **Table 1** shows communication overhead of N client requests to the server. The original protocol, as we know, communication overhead of N client requests to the server is 2N when the client initiates N requests. Compared to the original protocol, when the client initiates N requests, Two-dimensional table adds two session times but the other four schemes all add 2N; by this measure, Two-dimensional table outperforms the others.

The various computation costs of the server and client under all five schemes are reported in row 3 and row 4 in **Table 1**, respectively. Row 3 shows where if the server and client have communicated N times, Two-dimensional table needs Hash only once while Client puzzle scheme demands N times Hash. There are N times HMAC in the Cookie scheme. Stateless connections and Failing-together require several computation iterations (N times MAC/HMAC, 2N times Encryption, N times Decryption and 2N times Encryption, and N times Hash, respectively). In respect to server computation costs, Two-dimensional table again outperforms the others. The total computation costs of client of the proposed scheme and Cookie scheme are the lowest of the five, as reported in the fourth row of **Table 1**.

Table 1. Communication overhead of N client requests to the server

Index	**Proposed scheme**	**Stateless connections**	**Cookie**	**Failing-together**	**Client Puzzle**
Communication overhead	2	2n	2n	2n	2n
Computation costs of server	^{1}Hash	^{N}MAC(or HMAC), ^{2N}Encryption, ^{N}Decryption	^{N}HMAC	^{2N}Encryption, ^{N}Hash	^{N}Hash
Computation costs of client	None	^{N}Encryption, ^{N}Decryption	None	^{N}Encryption,^{N}Decryption, ^{N}Hash	^{N}PHC puzzle solution

Server storage costs	SN, RT_i, $HASH_i$, c[i]	K_a,K_b^{-1},K_{ba},K_{be}, K_{bm}	secret, K_{HMAC}	K_{ba},K_a,K_{be}	N_s,b
Client storage costs	RT_i, $HASH_i$	K_b,K_a^{-1},K_{be}	HMAC, context, IP-I	K_{ab},K_b,K_a,$^{-1}$,K_{be}	None

In row 5, Ns is a 64 bit nonce, as are the data block lengths of md5 and sha-1. The length of b is 8 bit (if b more than 8 bit, the PHC problem is unanswerable. The secret length of the Cookie is uncertain. To this effect, it is not possible to fully determine which scheme's server storage costs is the lowest. For the proposed scheme, in the worst case, NS: 64 bits, RTi: 64 bits, and HASHi: 64 bits.

The last row of **Table 1** shows where Client puzzle outperforms the others in terms of client storage costs, at no more than 128 bits (RTi: 64bit, HASHi: 64 bit). Cookie or Two-dimensional table could either come at the second-least client storage costs, but it is not possible to confirm which actually does as the length of the context is uncertain.

6. Conclusions

Rapid and extensive advancements in both the prevalence and capacity of intelligent devices such as smart phones have revolutionized the daily lives of many across the globe. Storage consumption is no longer a concern for most smart phone users, so the storage costs incurred by the server and client are not the primary concern among the six factors described above. Rather, communication efficiency and network security are the hallmarks of an effective (and attractive) smart device. The efficiency and overhead of communication are closely related to

computation costs, and there are countless links between communication efficiency and network security such as resource exhaustion-related DoS attacks.

We recommend communication overhead and security as crucial factors in the next generation of smart devices; computation costs to the server and client are also important considerations for the design of intelligent devices. Per the results of our six-factor analysis experiments, the Two-dimensional Table outperforms four other similar schemes in terms of communication overhead, security, and computation costs to the server and client.

References

[1] Cyrill Bannwart. Black Hat USA 2016 / DEF CON 24, September, 2016. Article (CrossRef Link).

[2] The State of the Internet [security] / Q1 2015, May, 2015. Article (CrossRef Link)

[3] The State of the Internet [security] / Q2 2017, August, 2017. Article (CrossRef Link)

[4] Blackscreen. DDoS 攻击的发展和应对, June, 2015. Article (CrossRef Link)

[5] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in *Proc. of the 2014 Network and Distributed System Security Symposium*, pp. 23-26, February, 2014. Article (CrossRef Link)

[6] P. Ferguson and D. Senie. "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," *BCP38 - RFC 2827*, May, 2000. Article (CrossRef Link)

[7] Anthony Sequeira, "BCP38 - RFC2827 Network Ingress Filtering: Defeat DoS with Forged Source Addresses," October, 2015. Article (CrossRef Link)

[8] Marc Kuhrer, Thomas Hupperich, Christian Rossow and Thorsten Holz, "Exit from Hell? Reducing the Impact of Amplification DDoS Attacks," in *Proc. of the 23rd USENIX Security Symposium*, pp. 111-125, August 20-22, 2014. Article (CrossRef Link)

[9] Seyed K. Fayaz, Yoshiaki Tobioka, Vyas Sekar and Michael Bailey, "Bohatei: Flexible and Elastic DDoS Defense,", in *Proc. of the 24th USENIX Security Symposium*, pp.817-832, August 12-14, 2015. Article (CrossRef Link)

[10] SSDP reflection DDoS attacks threat advisory, akamai's [state of the internet] / Threat Advisory, October, 2014. Article (CrossRef Link)

[11] IPv4 Multicast Address Space Registry, 2016. Article (CrossRef Link)

[12] IPv6 Multicast Address Space Registry, 2016. Article (CrossRef Link)

[13] Srinivas Arukonda and Samta Sinha, "The Innocent Perpetrators: Reflectors and Reflect ion Attacks," *ACSIJ Advances in Computer Science*, Vol. 4, No.13, pp. 94-98, January, 2015. Article (CrossRef Link)

[14] Tuomas Aura and Pekka Nikander, "Stateless connections," in *Proc. of International Conference on Information and Communications Security (ICICS'97)*, pp. 87-97, November 11-14, 1997. Article (CrossRef Link)

[15] K. Matsuura and H. Imai, "Protection of Authenticated Key-agreement Protocol against a Denial-of-Service Attack," in *Proc. of the International Symposium on Information Theory and Its Applications (ISITA'98)*, pp. 466-470, October, 1998. Article (CrossRef Link)

[16] L. Jiang, C. Xu, X. Wang and Y. Zhou, "Analysis and Comparison of the Network Security Protocol with DoS/DDoS Attack Resistance Performance," in *Proc. of High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conference on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on .IEEE*, pp. 1785-1790, August 24-26, 2015. Article (CrossRef Link)

[17] K. Matsuura and H. Imai, "Resolution of ISAKMP/Oakley key-agreement protocol resistant against Denial-of-Service attack," in *Proc. of Internet Workshop (IWS'99)*, pp. 17-24, February 18-20, 1999. Article (CrossRef Link)

[18] K. Matsuura and H. Imai, “Modification of Internet Key Exchange Resistant against Denial-of-Service,” in *Proc. of Internet Workshop 2000 (IWS 2000)*, pp.167-174, February, 2000. Article (CrossRef Link)

[19] V. Ragavi and G. Geetha, “Mitigating DoS Using Sensing Keys,” in *Proc. of Computing Sciences (ICCS), 2012 International Conference on .IEEE*, pp. 312-315, September 14-15, 2012. Article (CrossRef Link)

[20] Y. Zheng, “Digital signcryption or how to achieve cost (signature & encryption) << cost (signature) + cost (encryption),” in *Proc. of 17th Annual International Cryptology Conference Santa Barbara*, pp. 165-179, August 17-21, 1997. Article (CrossRef Link)

[21] Ak, Murat, Turgut Hanoymak, and Ali Aydın Selçuk, “IND-CCA secure encryption based on a Zheng–Seberry scheme,” *Journal of Computational and Applied Mathematics*, vol. 259, no.2, pp. 529-535, March, 2014. Article (CrossRef Link)

[22] C.P. Schnorr, “Efficient signature generation by smart cards,” *Journal of Cryptology*, vol. 4, no. 3, pp. 161-174, January, 1991. Article (CrossRef Link)

[23] Valluri, Maheswara Rao, "An identification protocol based on the twisted ring-root extraction problem," in *Proc of Industrial Control Systems Security (WCICSS), 2015 World Congress on. IEEE*, pp. 95-97, December 14-15, 2015. Article (CrossRef Link)

[24] Cynthia Dwork and Moni Naor, “Pricing via processing or combatting junk mail,” in *Proc. of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pp.139-147, August 16-20, 1992. Article (CrossRef Link)

[25] Ari Juels and John Brainard, “Client puzzles: A cryptographic countermeasure against connection depletion attacks,” *in Proc. 1999 Network and Distributed Systems Security Symposium (NDSS)*, pp. 151-165, February, 1999. Article (CrossRef Link)

[26] P.Karn and B.Simpson, “Photuris: Session Key Management Protocol,” IETF Network Working Group- RFC 2522, March, 1999. Article (CrossRef Link)

[27] Martin Abadi and Roger Needham, “Prudent Engineering Practice for Cryptographic Protocols,” *IEEE Transactions on Software Engineering*, vol. 22, no.1, pp. 6-15, January, 1996. Article (CrossRef Link)

Haiou Huang is studying for her Phd's degree in the College of Computer Science and Technology, Jilin University, Changchun. Her research interests include Data security and privacy, Wireless network.

Liang Hu has his BS degree on Computer Systems Harbin Institute of Technology in 1993 and his Ph.D. on Computer Software and Theory in 1999. Currently, he is the professor and Ph.D. supervisor of College of Computer Science and Technology, Jilin University, China. His main research interests include distributed systems, computer networks, communications technology and information security system, etc. As a person in charge or a principal participant, Dr Liang Hu has finished more than 20 national, provincial and ministerial level research projects of China.

Jianfen Chu received the M.S. and Ph.D. Degrees both from the College of Computer Science and Technology, Jilin University, Changchun. He is currently a sub-professor in the College of Computer Science and Technology, Jilin University. His research interests include Network Penetration, Data security and privacy.