Journal of Parallel and Distributed Computing 132 (2019) 150–159

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

A privacy preserved and credible network protocol

Zhongjiang Yao [a,b], Jingguo Ge [a,b,*], Yulei Wu [c,*], Linjie Jian [a,b]

[a] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China
[b] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, 100049, China
[c] College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4QF, UK

HIGHLIGHTS

- Privacy Protection for both the sender and the receiver.
- Avoiding the risk of privacy leak by a third-party proxy.
- New credibility metrics are proposed to determine how long to block the unwanted traffic.
- Good compatibility with the existing and future network architectures.

ARTICLE INFO

Article history:
Received 12 November 2018
Received in revised form 17 March 2019
Accepted 4 June 2019
Available online 13 June 2019

Keywords:
Privacy protection
Credibility
Accountability
Network protocol

ABSTRACT

The identities of packet senders and receivers are treated as important privacy information in communication networks. Any packet can be attributed to its sender for evaluating its credibility. Existing studies mainly rely on third-party agents that contain the packet sender's identity to ensure the sender's privacy preservation and credibility. In this case, packet senders run the risk that their privacy might be leaked by the agent. To this end, this paper proposes a Privacy Preserved and Credible Network Protocol (PCNP), which authorizes the agent to hide the identities of senders and receivers, while guaranteeing the credibility of a packet. The feasibility of the PCNP deployment is analyzed, and its performance is evaluated through extensive experiments.

1. Introduction

The research on privacy protection and credibility has made significant progress in the areas such as Internet-of-Things(IoT) [13,18,20,30], battery vehicles [15,16], big data [17,30], Blockchain [29], etc., but it has been relatively slow for the whole Internet. The current Internet architecture lacks security considerations in its initial design. With the rapid development of computer and communication technologies, the Internet becomes more complex, and its security is severely challenged. The pervasive use of the Internet has caused an incredible growth of unwanted traffic, such as spam, malware and malicious intrusions [32]. The Cybercrimes are constantly growing, and the global Cybersecurity market is expected to skyrocket to $231.94 billion by 2022 [28]. In terms of security, the packet owner (e.g., packet sender) always wants to show its credibility while keeping its privacy (i.e., the identity of packet sender) preserved [8]. For example, in a typical client–server (C/S) communication scenario, the server endeavors to have its identity hidden to avoid being attacked, while providing a credible service; on the other hand, the client expects to make the server trust its service request, without showing its identity. However, it is a great challenge for a sender and/or a receiver to be credible while keeping its privacy preserved.

Privacy preservation refers to the situation that no one else except the sender itself knows its identity [2,5,27], when a packet is issued out; no one else except the receiver itself knows its identity [7,26], when a packet is received; and no one can learn the payload except for the sender and the receiver [7]. All three types of network layer privacy data have been proposed in [23]. The packet sender and receiver are threatened with privacy leak due to the risks such as malicious network monitoring or attack.

Credibility indicates the authenticity and accountability of packet senders [9]. It is worth noting that it is different from the credibility of web page links [6]. In other words, the packet sender holds the responsibility for ensuring the accountability of the packet it provides. Anyone receiving the packet can verify its authenticity by challenging its source (i.e., packet sender) [1].

Existing researches mainly focus on either the privacy preservation or the credibility guarantee of packet senders, typically offering one by sacrificing the other. Recent studies, e.g., Accountable and Private Internet Protocol (APIP) [22] and Accountable and Private Network Architecture (APNA) [11], are to find a balance by making a tradeoff between these two factors. However,

* Corresponding author.
E-mail addresses: gejingguo@iie.ac.cn (J. Ge), y.l.wu@exeter.ac.uk (Y. Wu).

https://doi.org/10.1016/j.jpdc.2019.06.002
0743-7315/

they focused on using third-party agents to ensure the privacy preservation and the credibility guarantee for packet senders, where the sender's identity must be acquired by the agent. Under this circumstance, the packet senders run the risk that their privacy might be leaked by the agent.

To overcome this important problem, this paper proposes a Privacy Preserved and Credible Network Protocol (PCNP). The main contributions of this paper are summarized as follows:

- A Communication and Verification Agent (CVA) is proposed to ensure the privacy preservation of packet senders and receivers. Instead of using the sender's and receiver's identities in packet's source and destination fields, the CVA's identities are used. The packet transmission between senders and receivers thus looks like being forwarded between their associated CVAs.
- Considering the risk of privacy leak by the CVA, a temporary anonymous identity, *AmID*, is proposed. AmIDs are published to CVA for hiding sender's and receiver's identities, while CVA knows nothing about the owner of AmID. In other words, the sender can anonymously authorize the CVA to guarantee the credibility of a packet issued by it, without the risk of leaking its identity.
- New credibility metrics are proposed to measure the degree of the sender's credibility. The victim determines the blocking time of malicious traffic according to the credibility metrics.
- The deployment and security considerations of the proposed PCNP is discussed. Its performance is evaluated through extensive experiments. The results demonstrate the feasibility and merits of the proposed protocol.

The remainder of this paper is organized as follows. Section 2 outlines the adversary models for privacy preservation and credibility. Section 3 gives an overview of the proposed PCNP. The detailed design is discussed in Section 4. Section 5 analyzes its potential security issues. Section 6 presents the compatibility and deployment considerations of PCNP. Section 7 analyzes its performance. The related work is shown in Section 8. Finally, Section 9 makes a conclusion.

2. The motivation and adversary models

2.1. Motivation

Frequent Internet security incidents have resulted in serious problems for both the sender and the receiver of messages. Under this circumstance, the sender needs to pay more attention to its privacy protection and manage to minimize the information leakage (such as IP address, communication relationship, etc.) in the communication process; in addition, it promotes the recipient's need for receiving secure and reliable messages. Malicious traffic pursues the sender's responsibility, especially for the traffic that causes serious damage.

It is known from [12] and the TCP/IP network protocol stack that different privacy information needs to be protected at different levels. Privacy protection during network communication is usually to hide the user's node identity/location information. These information are used for packet routing, so the protection of these information belongs to the network level. Privacy protection at the application layer can only protect such as application type, data information and user activity patterns [14], and the application layer cannot verify the connection between the application layer part and the node identity/location. The node identity/location information is hidden at the data link layer, i.e., a MAC address, and there is little to do with the node identity/location information in the cross-network routing message.

In order to hide node identity/location information, various anonymous communication networks have been proposed at the network level, but they only have privacy protection capabilities and no accountability functions; technologies with accountability in network layer are rarely developed. The sender wants his/her privacy to be protected, while the receiver wants to receive packets that are authentic. In order to protect both senders and receivers, both privacy protection and accountability are indispensable. With the rapid development of network security techniques, how to balance privacy protection and accountability has become the most urgent issue.

2.2. Adversary models

2.2.1. Credibility consideration

The credibility refers to the authenticity of a packet which can be accounted to its sender. Anyone can trust a packet if it comes from a valid sender. In addition, receivers need to be able to challenge the sender if they receive unwanted traffic (i.e., malicious traffic or ping flood). When receiving unwanted traffic, the receiver needs to establish the relationship between the packet and its sender and take the corresponding actions to e.g. avoid the increased packet loss. To establish such relations, a third-party agent has to be introduced. The agent must not know the identity of the sender to be delegated but still can verify the authenticity of the delegated information.

The adversary model A perpetrator can initiate malicious actions without exposing its identity, to break the relationship between itself and its issued packets. Perpetrators may be in any locations in the network and can intercept, read, or counterfeit unencrypted network data.

2.2.2. Privacy consideration

Privacy preservation refers to the protection of the relationship between the packet and its sender, between the packet and its receiver, and between the sender and the receiver. To achieve the privacy preservation, the identities of senders and receivers must be hidden in the packet. Hiding identities of a sender and a receiver can also simply hide the communication between them. However, in the layer-2 broadcast domain, the sender or the receiver uses Medium Access Control (MAC) address to flood packets. The MAC may expose the privacy of the sender or the receiver, which is beyond the scope of this article. The identity of the sender or the receiver may also be leaked in higher layers, e.g., application layer, which is also beyond the scope of this paper.

The adversary model A perpetrator wants to break through privacy preservation to obtain the sender's identity from packets, which can help them find the sender's behavior. It is assumed that perpetrators can observe the packets transmitted over the network and also can compromise any network entities except for the sender and the public key management facility.

3. An overview of PCNP

3.1. The basic design of PCNP

PCNP is designed to provide the credibility of packets, while keeping the privacy preserved for packet senders and receivers. To describe the principle of the proposed PCNP, the key symbols used are shown in Table 1. In the proposed PCNP, a temporary identity called *AmID* is introduced for anonymous communication between the CVAs of the sender and the receiver. The AmID is used to reduce the risk of leaking sender's identity by the CVA, which will be detailed in Section 4. In order to ensure the packet to be delegated is credible, we need to redesign the packet formation, which is described in Section 3.1.2.

Table 1
The symbols used in the PCNP description.

Symbols	Description
SK_S, SK_D, SK_V	Symmetric keys shared by CVA and PKM, CVA and delegated host, CVA and routers, respectively
E_{SK}	Encrypted with symmetric key SK
E_{PKM}, E_C, E_S	Encrypted with public key of PKM/client/server
$seed_S$	Used to search symmetric key or identity of sender
SerID	The service identity in packet header, like port number
$Fp(D)$	The fingerprint of data D (i.e., packet, ID)
$H(D)$	The hash of data D
Certificate	The certification that server authorizes CVA to register service to name resolution server
M	Encrypted *Certificate* with the public key of PKM
C	The collection of AmID sent to CVA by receiver
$SIG_S(P)$	The signature of packet P signed by its source
Name	The name of a service
Exp	The symbol of deadline

3.1.1. Packet address format

The address format proposed in PCNP consists of two logical pieces: a Network IDentity (*NID*) identifies a network, which is used to forward packets to the destination network, and a Host IDentity (*HID*) identifies a host in the destination network. The complete address format can be expressed as *NID:HID*. Each host contains two types of *HID*: one is Real IDentity (*RID*), which is generated by using the public key that cannot be tampered and repudiated; the other is Anonymous IDentity (*AmID*) which is generated by using *RID*, MAC, random number and timestamp. *AmID* is used between CVA and packet sender or receiver (more details will be given in Section 4). *RID* is long-term effective, while *AmID* is temporary which is used only once.

NID:RID is a traditional address in a sense, identifying a host, which is globally unique and routable. However, *NID:AmID* is a temporary identity, but in the same format of *NID:RID*. To prevent the conflicts with the *AmID* generated by the other hosts in the same network domain, *AmID* should be registered with the Public Key Manager (PKM) for validity check. This check has to guarantee: (1) *AmID* is generated by the packet sender; (2) *AmID* is not being occupied. As a temporary identity, *AmID* has a *life-time* from the registration to the end of its usage. If its life-time is over, or it has been used, *AmID* will be removed from the *AmID* Cache (AC).

As a host may contact multiple hosts for a certain period of time or maintain multiple sessions with another host, it needs to generate multiple *AmID*s for privacy preservation. A network-layer protocol needs to implement an AC to save all AmIDs currently registered. When receiving an Address Resolution Protocol (ARP) packet, the host checks if the *HID* is *RID* first. If it is, the host responds with its MAC address. Otherwise, the host searches the AC. If the AC contains the *HID*, the host will return its MAC address; otherwise it ignores the ARP.

3.1.2. Packet format

To hide the identities of sender and receiver, PCNP replaces the source address, in traditional packet header, with the Verify and Receive Identity (VRI) of source and replaces the destination address with destination VRI, as shown in Fig. 1. The source VRI is used to verify the authenticity of packets, and the destination VRI is used to forward the packet to its receiver. Rather than identifying the sender and the receiver, the source VRI identifies the CVA who vouches for the packets, and the destination VRI identifies the CVA who delegates the received packets. If the communicators are not anonymous, the VRIs are filled with the

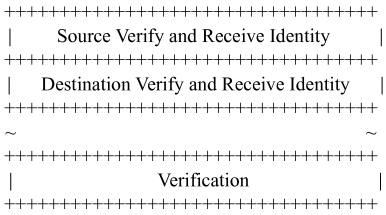


Fig. 1. The Packet Format in PCNP.

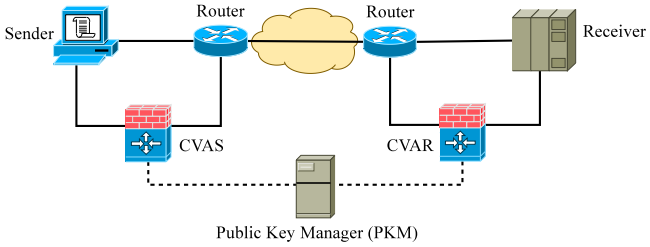


Fig. 2. An Overview of PCNP.

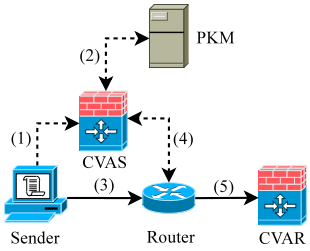


Fig. 3. The sending process: (1) Sender sends *Brief* to CVAS; (2) CVAS verifies the authenticity of the *Brief*; (3) Sender sends packet; (4) Router verifies the authenticity of the packet; (5) Router forwards the packet.

NID:RID of the sender and the receiver. A Verification field (Ver) is introduced for two roles: one is used to verify the authenticity of packets, and the other is used to challenge the packet sender [4]. The Ver consists of $seed_S$ and the signed fingerprint of payload $SIG_S(Fp(P))$. The $seed_S$ is encrypted with Exp_S by the public key of PKM which is shared by sender and PKM. The $seed_S$ is used to ensure the validity of the packet, and the $SIG_S(Fp(P))$ is mainly used to authenticate the identity of the sender of the packet.

$$[E_{PKM}(seed_S, Exp_S), SIG_S(Fp(P))] \quad (1)$$

$seed_S$, which is refreshed periodically, is a secret token generated by the sender. If a perpetrator generates a public key conflicting with that of an existing host, the packets sent by the perpetrator will be failed for verification, as the perpetrator knows nothing about the $seed_S$.

3.1.3. Communication

The overview of PCNP is shown in Fig. 2. The Sender's CVA (CVAS) and the Receiver's CVA (CVAR) are chosen by the sender and the receiver, respectively, which are used to hide their identities. PKM is the facility used to manage public keys and provide authentication, which prevents host from generating public keys randomly. To describe the PCNP clearly, the complete communication process is divided into three parts: sending process, receiving process, and accountability process.

(1) Sending process. As shown in Fig. 3, to keep the credibility of a packet while hiding the sender's identity, CVAS is introduced to establish the relationship between packet and its sender. Before sending a packet P, the sender will anonymously send a

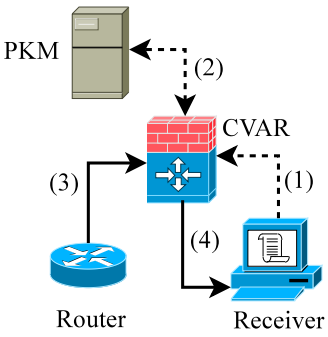


Fig. 4. The receiving process: (1) Receiver sends a *Publish* to CVAR; (2) CVAR verifies the authenticity of *Publish*; (3) Router forwards a packet to CVAR; (4) CVAR forwards the packet to receiver after being processed.

Brief, which is a packet carrying the summary of *P*, to the CVAS with *AmID*. For the authenticity of *Brief*, CVAS needs to verify its authenticity from PKM. At the same time, the sender sends out *P*. Due to the lack of sender's identity, the routers on the forwarding path will challenge the credibility of *P*. To do this, routers can send *Verify*, which is the packet used for verifying the authenticity of *P*, to the CVAS which caches the summary of *P* (more details will be provided in Section 4). If every packet of a stream gets verified, the forwarding performance will be seriously affected (i.e., the speed of packet transition will be slowed down, and network congestion may be produced). Therefore, a local whitelist is introduced to cache the summary of packet that has passed the verification.

(2) Receiving process. A receiver can be a server which provides certain services (e.g., website and cloud computing) or a host which receives a response. As shown in Fig. 4, if it is a server, it will anonymously send *Publish* to CVAR with *AmID* to hide its identity (more details are shown in Section 4). A *Publish* is a packet used to authorize its CVAR to register a service to Domain Name System (DNS) with the *NID:RID* of CVAR. Due to the lack of the receiver's identity in the *Publish*, CVAR needs to verify the authenticity of *Publish*. When requesting a service of receiver, the sender issues a request *P* to CVAR. CVAR forwards *P* to the receiver with *AmID*. If it is the host to receive a response, CVAR receives the response packet and forwards it to the receiver with *AmID*.

(3) Accountability process. As shown in Fig. 5, when receiving unwanted traffic (e.g., malicious traffic), the receiver sends *Shutoff* to decrease the resulting negative impact. The *Shutoff* is the packet used to report to CVAS and block unwanted traffic quickly. During the forwarding of *Shutoff*, routers can verify it as a standard packet as mentioned in the Sending process. When CVAS receives a *Shutoff*, it finds the related records and stops providing the verification of related packets, and finally forwards the encrypted *Shutoff* to the sender. The sender should then stop sending the traffic. A malicious host may ignore the *Shutoff* and the negative impact can continue spreading. But the CVAS can fail the packet verification and stops the traffic. To achieve better performance, the router introduces a Blacklist, which caches the blocked packets.

The proposed PCNP has four advantages. First, CVA protects the privacy of senders, although CVAS is not clear to whom it delegates. Second, it protects the privacy of receivers, although CVAR is not clear of those delegated receivers. Third, PCNP hides the communication relationship between sender and receiver. Finally, it provides a credible communication while protecting the privacy of senders and receivers.

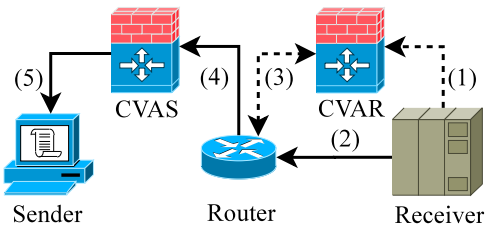


Fig. 5. The accountability process: (1) Receiver sends the *Brief* of *Shutoff*; (2) Receiver sends *Shutoff*; (3) Router verifies the *Shutoff*; (4) Router forwards the *Shutoff* to CVAS; (5) CVAS forwards the processed *Shutoff* to sender.

4. The details of PCNP

This section shows the detailed design of PCNP, where the design of anonymity method, key management and other mechanisms are inspired by [33] in Chapter 15. It should be noted that all packets in PCNP contain *Ver*. For the sake of the clarity of illustration, the *Ver* is not included in describing a packet in this section.

4.1. Communication and verification agent (CVA)

CVA plays a key role in the proposed PCNP. It proves the relationship between packet and its sender/its receiver.

4.1.1. The relationship between packet and its sender

To prove the relationship between the packet *P* and its sender, the sender issues a *Brief* with *NID:AmID* to CVAS before sending *P*, which can be used as digital certificate if legal intervention is required. The *Brief* contains the 5-tuple (i.e., source VRI, source SerID, protocol type, destination SerID, and destination VRI) and the fingerprint of *P*. SerID is selected from the SerID collection, which is negotiated between CVA and the delegated host. To prevent the anonymity scope of senders from being narrowed by others using the link between 5-tuple and *NID*, we suggest that the payload of *Brief* should be encrypted using symmetric key SK_D. SK_D is negotiated after the sender establishes a connection with CVAS using *AmID*, as shown in Fig. 6. When receiving a *Brief*, CVAS needs to verify its authenticity from PKM (more details are given in Section 4.2.1).

The sender sending *Brief* to CVAS can be expressed as:

$$Brief(E_{SK_D}(5-tuple, SIG_S(Fp(P)), AmID)) \quad (2)$$

4.1.2. Forward packets to the receiver

The basic function of CVAR is to forward packets to receivers. As mentioned in Section 3, receivers can be divided into two types: a server (i.e., web sites, cloud computing), from which a sender requests a service; and a host, who receives responses. If the receiver is a server, it authorizes CVAR to register its service, which will be detailed in Section 1.

(1) Publish service.

Publish The receiver sends *Publish* to authorize CVAR to register services with the *NID:RID* of CVAR. *Publish* is a packet for authorizing, which contains the service name, encrypted authorization Certificate, and a collection of *AmIDs*. The *Certificate* is the credential that a server authorizes CVAR, which has been uploaded and confirmed by PKM. The *Certificate* contains the service name, the identity of server, and the identity of CVAR to be authorized. For the credibility of *Publish*, the service name, encrypted *Certificate*, and the hash of collection of *AmIDs* will be sent to PKM for verification (more details will be available in Section 4.2). The *AmID* collection is used for CVAR to forward

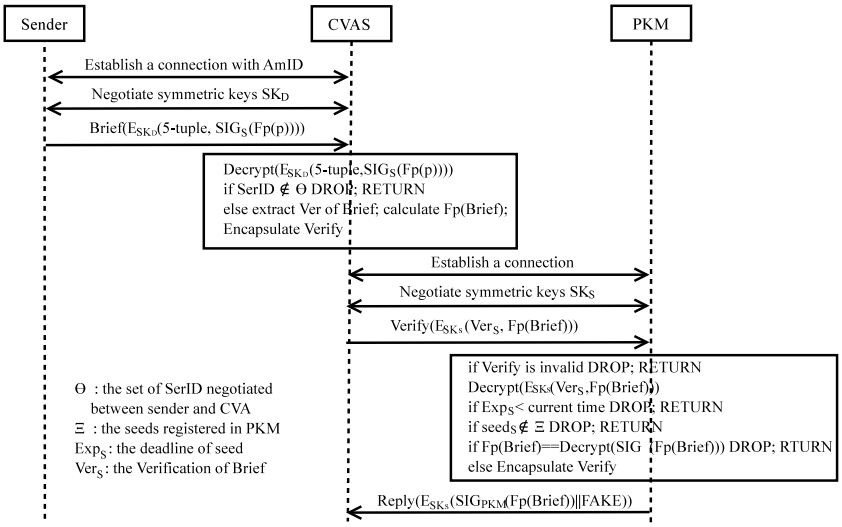


Fig. 6. The process of *Brief*.

requests with *NID* to the server as the destination VRI. Each service has an *AmID* collection, and each *AmID* is assigned to a request. The payload of *Publish* is encrypted with S_D which is shared between CVAR and server.

The receiver authorizing CVAR to register service can be shown as:

$$Publish(E_{SK_D}(Name, M, C)) \tag{3}$$

Update The size of AmID collection is limited and needs to be updated periodically. This paper proposes two options for updating: *Interval-based update* and *Quantity-based update*.

Interval-based Update. Considering that some servers may have less traffic, *AmID* collection may have not been used up in updating interval, *interval*. As the *AmID* runs the risk of being leaked, and the sender loses its anonymity beyond the update interval, the *AmID* collection is still updated periodically. Therefore, each server has a *Timer*. When there is time T left before the *interval* ends ($interval < life\text{-}time$), the server starts to generate a new *AmID* and updates them to CVAR when time τ ($\tau < T$) is left. The update is anonymous, so it needs to be verified similar to the *Publish*.

Quantity-based Update. Considering that some servers may have amounts of traffic, and the *AmID*s in *life-time* may be used up, PCNP provides a quantity-based update mechanism. Each server sets a Counter to record the number of remaining *AmID*s. When Counter reaches the threshold θ ($\theta \geq 1$), the server updates *AmID*s to CVAR. The update will be handled in the same way as the Interval-based scheme.

(2) Forward packet. The destination VRI in the packet is a CVAR's NID:RID, which is not the final receiver. CVAR should forward packets to its final receiver.

To avoid observer's tracking, CVAR establishes a new connection with delegated receiver using a new *AmID*. If the packet is a request to a server, *AmID* is selected from the *AmID* collection; if the packet is a response, *AmID* comes from a *Brief* record. Then, CVAR and the delegated receiver negotiate a symmetric key SK_D. The CVAR re-encapsulates the packet using SK_D.

4.2. Credibility measures

Credibility measures are necessary in PCNP, mainly including two aspects: (1) Verification and (2) Accountability.

4.2.1. Verification

It is dangerous that a CVA knows nothing about to whom it delegates. For credibility, a packet needs to be verified for its authenticity. This section makes a summary of the verification in PCNP as follows.

(1) CVA verifies the authenticity of brief. CVA verifies the authenticity of *Brief*. The foundation for providing the verification of packet is the authenticity of *Brief* in CVAS. Specifically, the CVAS establishes a connection with PKM and negotiates a symmetric key SK_S. The CVAS encrypts the Ver of *Brief* with SK_S and encapsulates it into *Verify*. Then, the CVAS sends the *Verify* to PKM for verification. The PKM mainly verifies two things: (1) check the Ver of *Verify* for its authenticity; (2) check the Ver of *Brief* contained in *Verify*. For the check, it mainly contains that the seed in Ver has been registered in PKM, the signature is genuine, and the fingerprint matches the fingerprint re-calculated by CVAS. The details are shown in Fig. 6.

(2) CVA verifies the credibility of publish. The foundation for providing the credible authorization of a service is the authenticity of *Publish* received by CVAR. Specifically, the CVAR establishes a connection with PKM and negotiates a symmetric key SK_S. The CVAR encrypts the Ver of *Publish* with SK_S and encapsulates it into *Verify*. Then, the CVAR sends the *Verify* to PKM for verification. The PKM mainly verifies four things: (1) the *Verify* is valid; (2) the Ver of *Publish* is valid; (3) the *Certificate* matches with a registered one in PKM; and (4) the $H(C)$ matches with the hash of the registered *AmID* collection. If all the conditions are satisfied, PKM returns the fingerprint of *Publish* signed by PKM, and CVAR registers the service to DNS. The details are shown in Fig. 7.

(3) Routers verify authenticity of packet to be forwarded. When receiving a packet, routers check their Whitelist in which the packets have been verified. If it is not in, routers search the Blacklist in which the packet is blocked. If it is blocked, routers drop the packet directly. When the packet is neither in Whitelist nor in Blacklist, routers send *Verify* to verify the packet authenticity (see Fig. 8). If the packet is credible, routers continue forwarding it.

4.2.2. Accountability

A receiver should only account for unwanted packets for any damages.

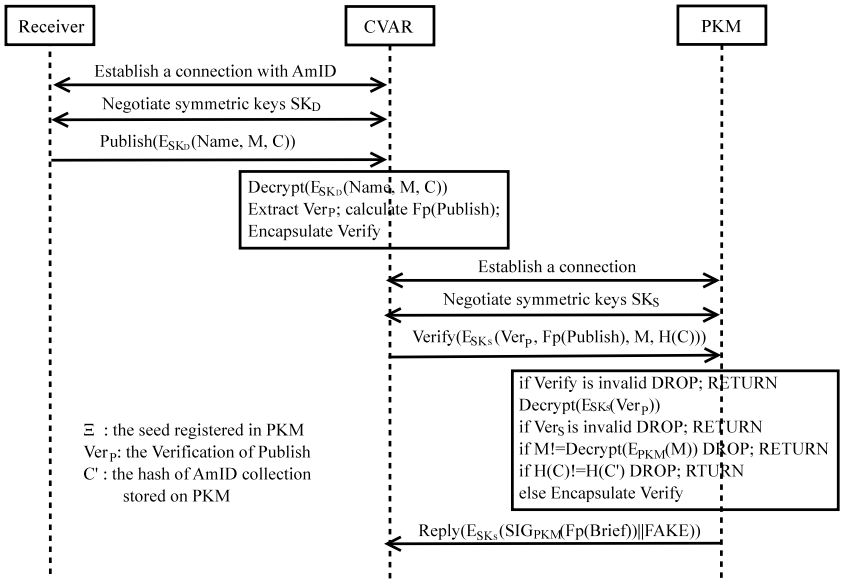


Fig. 7. The verification of *Publish*.

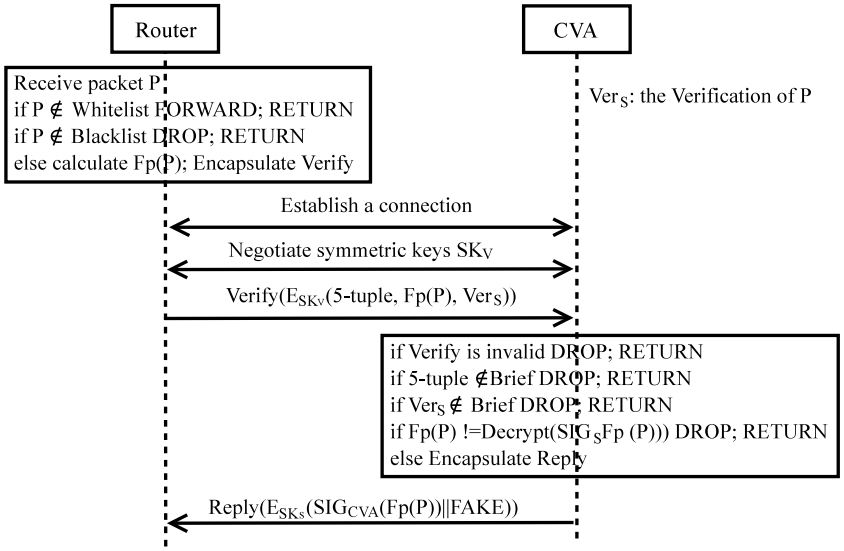


Fig. 8. The verification of a packet.

(1) Shutoff. *Shutoff* is used to block malicious traffic of the sender. However, the malicious sender may ignore the Shutoff. Therefore, we use routers and CVA to play a supporting role. Once receiving a *Shutoff*, routers search the information carried in the *Shutoff* in the local Whitelist. If it is not in, routers put it in their Blacklist; if it is in, routers confirm the *Shutoff* from the CVAR. If the *Shutoff* is valid, it will be removed from Whitelist and added into Blacklist; otherwise routers ignore the *Shutoff*. Once being found in Blacklist, the packet will be discarded.

Granularity The traditional way uses the 5-tuple in packet header to form Flow Identity. The SerID is one component of 5-tuple in PCNP like port number. When several senders use the same proxy, it is difficult to distinguish those access to the same service on the same server for the 5-tuple, if they use the same source SerID, destination SerID and VRI. To provide finer-granularity traffic accountability, CVA and sender (or receiver) negotiate SerID collection with *AmID*. Different traffic will have different SerIDs: a sender can use a SerID to a host, and it can also use a SerID to a session, even to a packet.

Credibility metrics In APIP, long-term fix and short-term fix are proposed [22], which lacks flexibility. Therefore, we propose new credibility metrics: the Damage Degree (DD) ψ, the Historical Safety Assessment (HSA) Φ, and Benign Ratio (BR). DD is calculated according to resource consumption, performance influence, and data damage. HSA obeys a power function with a base of mean value of historical Ψ, multiplied by a constant coefficient to make the value range between [0, 1], which represents the reputation of a node. BR is the ratio of the benign packets δ and total volume of packets Σ. DD, HSA and BR are typical metrics that can be used to evaluate sender reputation, and they are the basis for determining the blocking time. The higher the DD and the lower the HSA, the longer the blocking time by *Shutoff* is. When receiving unwanted traffic, the receiver determines the blocking time Γ for the sender to fix the problems, according to the credibility metrics which are computed in PKM as follows.

$$\Gamma = (\psi + (\frac{\delta}{\Sigma} \cdot \Phi)^{-1}) \cdot \Omega, \mu < \frac{\delta}{\Sigma} \cdot \Phi < 1, 1 \leq \psi \leq 10 \qquad (4)$$

where Ω is the time-unit of blocking time. We evaluate the sender's reputation by calculating $\frac{\delta}{\Sigma} \cdot \Phi$ and set a threshold μ as a criterion for evaluating sender reputation. When the reputation is lower than μ, all traffic of the sender is permanently blocked; otherwise, the blocking time is calculated according to the above formula.

Security NIC Perpetrators usually do not admit their malicious actions and even ignore *Shutoff*. We therefore introduce a Security Network Interface Card (S-NIC). S-NIC is assembled on the host using hot swapping and burned asymmetric key and identity. Host cannot modify them via Operating System (OS) and malicious programs. S-NIC's built-in cache space is used to cache fingerprint of packets sent through it. S-NIC checks the Ver of the packet sent out through this S-NIC. Once receiving a *Shutoff*, OS stops the application, and S-NIC filters related packets.

5. Security analysis

5.1. Bootstrap trust

Although PCNP uses public key to generate self-certifying identity (*RID*), additional supports are required for credibility. *Brief* contains the sender's signature to prove its authentication and encrypted with symmetric key for the privacy of senders. Except for those mentioned in *Brief*, *Publish* carries an encrypted credential for authorizing CVA to register service. *Verify* includes two categories to keep the credibility: first, CVA verifies the authenticity of delegated information; second, anyone can verify the authenticity of packets from the CVA. The payload of *Verify* is also encrypted with symmetric key. *Shutoff* contains Ver. If there are any doubts about a *Shutoff*, the sender can verify it from CVA, who caches the *Brief* of the *Shutoff*.

5.2. Attack on credibility

5.2.1. Spoofing

Perpetrators have two possible spoofing packets. First, intercepting a packet and reinjecting it into network for replay attack. Records in Whitelist have a lifetime. If they are outdated when replay attack happens, the verification will fail; if they are within time limits, the receiver sends *Shutoff*. Second, perpetrators generate public key and identity randomly. If the public key generated has not been registered in PKM, the verification of the *Brief* signed by this public key will be failed, and the CVAS will not store its record. If the public key has registered in PKM, the *Brief* verification will be failure as the perpetrator knows nothing about the secret token.

5.2.2. Risk of CVA

If perpetrators send *Brief* with false *HID*, the verification will be failed as the seed cannot be known by perpetrators. Perpetrators can also send a large amount of false *Brief* or false packets causing routers to send a lot of *Verify* which may lead DoS attack and make CVA lose service capability. Although CVA has higher reputation, it still can be compromised. A compromised CVA may fail a valid packet or authenticate an invalid packet. In this case, it is worth noting that the CVA can be replaced freely when sender or receiver detects these malicious actions, because the correspondence between CVA and server is not fixed.

5.3. Attack on privacy

The goal of privacy preservation is to protect the relationship between packet and its sender (or receiver), the communication relationship between sender and receiver. In APIP [22] and APNA [11], the privacy can be available as long as the agent is compromised. While in PCNP, all records in CVA, which is the agent containing only *AmID*, do not contain sender's or receiver's identity, which means the privacy mentioned above cannot be leaked. Though *AmID* can be obtained via CVA, it is not sure to whom *AmID* belongs.

6. Deployment and compatibility

6.1. Deployment

When deploying PCNP, a highly trusted third-party needs to be selected as a CVA (e.g., ISP, which is short for Internet Service Provider, or a company with high reputation). In current network, Network Address Translation (NAT) and Border Router (BR) enjoy good reputations. When receiving a packet, CVA replaces the destination address with *AmID*, which is similar to NAT. When a packet leaves its source domain, NAT can provide verification and accountability for the packet through it. In the Autonomous System (AS), the BR, which is the AS gateway for packet transition, can record all packets through it and prove the authenticity of packets within it. That is to say, both NAT and BR can be used as CVAs.

6.2. Compatibility

If PCNP is introduced into the widely used TCP/IP network, the works below are needed: deploy CVAs; add additional functions to PKI; use source and destination addresses in packet header as source and destination VRI; and fill Ver in optional field of IP header. It is worth noting that the NAT or BR can be configured as the CVA. In order to ensure IP addresses are credible and comply with PCNP, PKI needs to add seed, authorization credential and verification function. In a Local Area Network (LAN), many IP addresses with the same *NID* must be reserved as *NID:AmID*. ISPs can dynamically allocate IP addresses as *NID:AmID* by DHCP. IPv6 is considered as the next generation Internet protocol and possesses a lot of similarity with IPv4. IPv6 can be tuned in the same way as IPv4 when PCNP is introduced. In addition, some well-known projects on future Internet architecture, e.g., MobilityFirst [25] and HIP [21], have investigated the way of adapting to the changes of the future traffic types. MobilityFirst and HIP introduced the flat naming scheme. MobilityFirst separates host identity from location with Globally Unique Identifier (GUID). PCNP only concerns how to hide GUID without worrying about Network Address (NA). In order to use the proposed PCNP in MobilityFirst, we need to deploy CVAs, replace source and destination GUIDs in packet with the GUIDs of CVAS and CVAR, and add Verification in Service Header Extensions. As HIP packet header is IPv6 header, it can follow IPv6 adjustment method to use PCNP.

7. Evaluation and performance analysis

To evaluate the proposed PCNP, this paper uses IP as overlay. The sender, router and receiver are implemented with C++ for faster packet forwarding; the CVA and PKM are implemented with JAVA for multi-platform compatibility. The experiment is made on two physical computers: the OS is Ubuntu 16.04 whose core is 4.4.0-83-generic, and CPU is i5-3470T @ 2.9 GHz *4. The sender, router and receiver run as processes on one computer, while CVA and PKM run as processes on the other one.

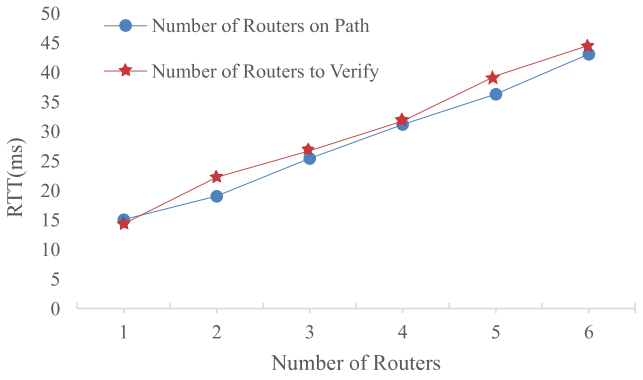


Fig. 9. The RTT of a communication between the sender and the receiver.

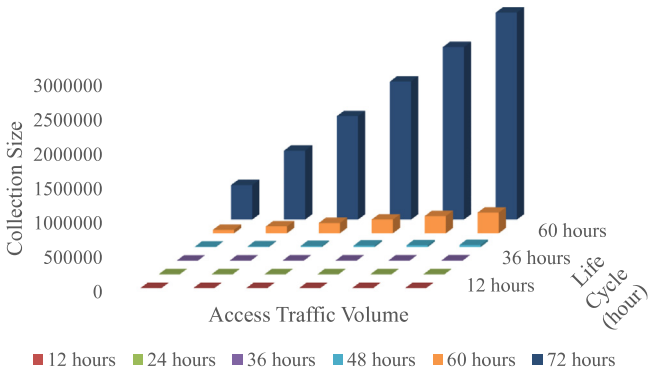


Fig. 10. The relationship between *AmID* Collection and *AmID*'s lifecycle, access traffic volume.

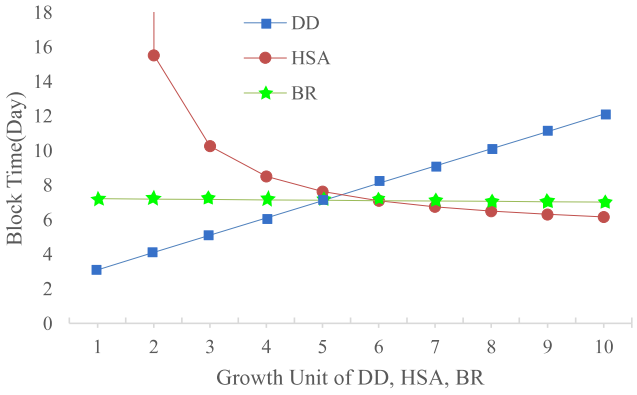


Fig. 11. The factor influent block period.

7.1. Round-Trip Time (RTT)

RTT is the time from sending a request by the sender to receiving its response. It directly affects how users feel the network they used subject to a given network protocol and is one of the most important indicators for evaluating the feasibility of a network protocol. In this experiment, we change two variables: the number of routers on the forwarding path and the number of routers verifying the authenticity of packet. Fig. 9 depicts the RTT between sender and receiver, whose *x-axis* represents the number of routers and the *y-axis* denotes the RTT. For each additional router forwarding on the routing path without verification as what the Internet Protocol (IP) does, one link transmission time T_t and one router processing time T_p are required. The total time is $T_{sum} = \sum_{i=h}^{I}(N \cdot (T_t^i + T_p^i) + T_p^i)$,where I is the total number of packets. From this figure, we can see that the RTT is increasing with the increase in the number of routers on the forwarding path or the number of routers providing verification. It takes longer time for RTT to go through routers with verification than routers only forwarding packets. But the time used for verification does not obviously increase with the number of by-passing routers. The ith packet verification time required for the router is t^i. The total time with verification is $T_{sum} = \sum_{i=h}^{I}(N \cdot (T_t^i + T_p^i + t^i) + T_p^i)$. However, due to the introduction of whitelist and the blacklist, only the first one packet needs to be authenticated by the routers of the path, and the subsequent packets only need to access the blacklist and whitelist of the routers. This time for accessing the blacklist and the whitelist is much shorter than the packet verification time, and then can be ignored. Thus, the total time with verification also can be expressed as $T_{sum} = \sum_{i=h}^{I}(N \cdot (T_t^i + T_p^i) + T_p^i)$. Therefore, the RTT is acceptable in reality as the value is millisecond.

7.2. AmID collection

AmID Collection is an important part of PCNP privacy protection. The frequency of *AmID* Collection update and the size of *AmID* Collection not only affect PCNP's ability to protect privacy but also affect network congestion. The relationship among the size of *AmID* collection, lifecycle and traffic needs to be taken into consideration. We use RSA1024 and SHA256 to generate *HID*. Assuming that the traffic volume changes between 10 and 1,000,000, and the lifecycle varies from 12 to 72 h. Fig. 10 depicts the space usage of CVAR, whose *x-axis* is the access traffic volume that represents the volume of received packets by the receiver, the *y-axis* denotes the lifecycle, and the *z-axis* represents the space usage. From Fig. 10, we can see when traffic is constant, *AmID* collection size is linearly positive correlated to lifecycle. When lifecycle is constant, the size of *AmID* collection is linearly positive correlated to traffic.

When the access traffic volume is constant, the longer the *AmID* Collection update cycle is, the more *AmID*s needs to be included in the *AmID* Collection in order to ensure normal service delivery, but the longer update cycle also increases the risk of *AmID* being leaked. When the *AmID* Collection has a fixed capacity, the larger the volume of access is, the smaller the update cycle is required by the *AmID* Collection, but the risk of *AmID* being leaked is reduced. When the *AmID* Collection update period is fixed, the *AmID* Collection update cycle will be shortened in order to maintain normal service capabilities.

7.3. Blocking time

Blocking unwanted traffic is a specific punishment for PCNP to account for malicious users, and is a direct indicator that affects the credibility of PCNP. DD is divided into 10 grades according to the degree of damage. BR changes between 0.9 and 1, and μ=0.9. HSA varies between 0 and 1. The blocking time is the period the traffic to be blocked. The Growth Unit of DD, HSA and BR are set to be 1, 0.1 and 0.01, respectively. Fig. 11 depicts the Block Time, whose *x-axis* represents the growth unit of DD, HSA and BR, and the *y-axis* denotes the blocking time. From this figure, we can see that DD has a linear positive correlation with blocking time. For the HSA, the lower the value, the greater the impact on blocking time. The higher DD and HSA, the blocking time is less affected by the BR.

The effect of the HSA on the blocking time follows a power function with a parameter of -1, which means the lower a sender's historical reputation is, the more drastic effect will be

exerted on the blocking time. Because the low historical reputation indicates that the sender is more frequently producing network anomalies, it will always be the object of vigilance, and the punishment should be heavier. If the history reputation is high (there is rarely a network anomaly in history), the influence of HSA on blocking time is lighter, while DD plays a decisive role in the traffic accountability. The larger the DD is, the more serious the damage will be caused by the unwanted traffic, and therefore more timely measures to stop the loss are needed.

8. Related work

As the existing studies may consider credibility or privacy preservation, for explicit illustration, we separate this section into two parts: Credibility and Privacy.

8.1. Credibility

Although observers can tell the identities of senders and receivers of a packet in TCP/IP network, IP address is not credible. In order to solve IP address credibility issues, S-BGP [10] added address attestation field and introduced two PKI infrastructures to bind public key with IP address and AS. To achieve more credibility, AIP [3] made public key as host identity for the first time. However, TCP/IP and AIP had no online accountable measures and cannot protect privacy. APIP balanced accountability and privacy in [22] by introducing accountability agent. APNA proposed a privacy preservation scheme with ISP assistance in [11]. It introduced temporary EphIDs to support credibility and privacy preservation. In APIP and APNA, their credibility is mainly based on the delegate or ISP. In [31], the privacy preservation trust management system can only detect and control unwanted traffic without holding any measures accountable.

8.2. Privacy

Mix is an anonymous approach proposed by *Chaum* in [7]. The essence of Mix is a middleman mixing multiuser messages, confusing the observer's sight. Onion routing [26] algorithm uses source routing to randomly select three onion routers to form a transmission path and encrypt data for many times to effectively hide sender and receiver. However, the receiver cannot account for malicious traffic. AHP asks for ISP's assistance and uses NAT to map IP address to hide IP and stick IP to achieve privacy in [24]. APIP [22] separated the accountability role from the source address. APNA [11] used Management Service (MS) to issue an EphID, which is used to replace sender address. However, those studies do not provide destination host privacy preservation, and the third-party runs the risk of leaking sender's privacy. Based on AIP and APIP, [19] proposed a new architecture for trusted anonymous use of services in distributed computing networks.

9. Conclusion

This paper has tackled the latest research on credibility and privacy preservation in network layer and proposed PCNP. PCNP has introduced CVA to hide communicator's identity and performed trusted communication under the condition that communicators are anonymous to CVA. The security, deployment and compatibility of the proposed PCNP has been analyzed. The performance of PCNP has been validated by extensive experiments. The results have demonstrated the feasibility of the proposed PCNP.

Acknowledgments

This work is partially supported by the National Science and Technology Major Project of the Ministry of Science and Technology of China (Grant No.: 2017ZX03001019) and the Science and Technology Service Network Initiative (STS) Project of Chinese Academy of Science (Grant No.: Y7X0071105).

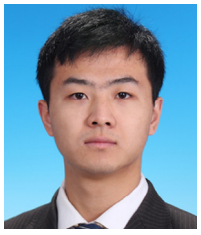
Declaration of competing interest

No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.jpdc.2019.06.002.

References

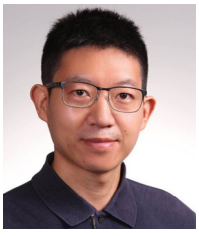
[1] M. Afanasyev, T. Kohno, J. Ma, N. Murphy, S. Savage, A. Snoeren, G. Voelker, Privacy-preserving network forensics, Commun. ACM 54 (5) (2011) 78–87, http://dx.doi.org/10.1145/1941487.1941508.
[2] N. Ahmad, H. Cruickshank, Y. Cao, F. Khan, M. Asif, A. Ahmad, G. Jeon, Privacy by architecture pseudonym framework for delay tolerant network, Future Gener. Comput. Syst. (2017) http://dx.doi.org/10.1016/j.future.2017.11.017.
[3] D. Andersen, H. Balakrishnan, H. Feamster, T. Koponen, D. Moon, S. Shenker, Accountable internet protocol (aip), SIGCOMM Comput. Commun. Rev. 38 (4) (2008) 339–350, http://dx.doi.org/10.1145/1402946.1402997.
[4] S. Bechtold, A. Perrig, Accountability in future internet architectures, Commun. ACM 57 (9) (2014) 21–23, http://dx.doi.org/10.1145/2644146.
[5] J. Boyan, The anonymizer - protecting user privacy on the web, 1997.
[6] J. Caverlee, L. Liu, Countering web spam with credibility-based link analysis, in: Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC '07, 2007, pp. 157–166, http://dx.doi.org/10.1145/1281100.1281124.
[7] D. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, Commun. ACM 24 (2) (1981) 84–90, http://dx.doi.org/10.1145/358549.358563.
[8] P. Fischer, P. Stocken, Imperfect information and credible communication, J. Account. Res. 39 (1) (2010) 119–134.
[9] C. Gao, N. Iwane, A social network model for big data privacy preserving and accountability assurance, in: Consumer Commun. and NETWORKING Conf. IEEE, 2015, pp. 9–12, http://dx.doi.org/10.1109/CCNC.2015.7157940.
[10] S. Kent, C. Lynn, K. Seo, Secure border gateway protocol (S-BGP), IEEE J. Selected Areas Commun. 18 (4) (2000) 582–592, http://dx.doi.org/10.1109/49.839934.
[11] T. Lee, C. Pappas, D. Barrera, P. Szalachowski, A. Perrig, Source accountability with domain-brokered privacy, in: Proc. of the 12th Inter. on Conf. on Emerging Networking EXperiments and Tech., vol. 44, no. 4, 2016, pp. 345–358, http://dx.doi.org/10.1145/2999572.2999581.
[12] S. Li, T. Tryfonas, H. Li, The internet of things: a security point of view, Internet Res. 26 (2) (2016) 337–359.
[13] S. Li, S. Zhao, P. Yang, P. Andriotis, L. Xu, Q. Sun, Distributed consensus algorithm for events detection in cyber physical systems, IEEE IoT J. (2019) http://dx.doi.org/10.1109/JIOT.2019.2906157.
[14] S. Li, S. Zhao, Y. Yuan, Q. Sun, K. Zhang, Dynamic security risk evaluation via hybrid Bayesian risk graph in cyber-physical social systems, IEEE Trans. Comput. Soc. Syst. 5 (4) (2018) 1133–1141.
[15] H. Liu, H. Ning, Y. Zhang, Q. Xiong, L.T. Yang, Role-dependent privacy preservation for secure v2g networks in the smart grid, IEEE Trans. Inf. Forensics Secur. 9 (2) (2014) 208–220.
[16] H. Liu, H. Ning, Y. Zhang, L.T. Yang, Aggregated-proofs based privacy-preserving authentication for v2g networks in the smart grid, IEEE Trans. Smart Grid 3 (4) (2012) 1722–1733.
[17] Q. Liu, A. Srinivasan, J. Hu, G. Wang, Preface: Security and privacy in big data clouds, Future Gener. Comput. Syst. 72 (2017) 206–207, http://dx.doi.org/10.1016/j.future.2017.03.033.
[18] J. Lopez, R. Rios, F. Bao, G. Wang, Evolving privacy: From sensors to the internet of things, Future Gener. Comput. Syst. 75 (2017) 46–57, http://dx.doi.org/10.1016/j.future.2017.04.045.
[19] Y. Ma, Y. Wu, J. Ge, J. Li, A new architecture for anonymous use of services in distributed computing networks, in: 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications, ISPA/IUCC, 2017, pp. 368–374, http://dx.doi.org/10.1109/ISPA/IUCC.2017.00059.
[20] Y. Ma, Y. Wu, J. Ge, J. Li, An architecture for accountable anonymous access in the internet-of-things network, IEEE Access 6 (2018) 14451–14461, http://dx.doi.org/10.1109/ACCESS.2018.2806483.

[21] R. Moskowitz, P. Nikander, Host identity protocol (HIP) architecture, RFC 4423 (2006) 1–24, http://dx.doi.org/10.17487/rfc4423.
[22] D. Naylor, M. Mukerjee, P. Steenkiste, Balancing accountability and privacy in the network, SIGCOMM Comput. Commun. Rev. 44 (4) (2014) 75–86, http://dx.doi.org/10.1145/2740070.2626306.
[23] R. Oppliger, Privacy protection and anonymity services for the World Wide Web (WWW), 16 (4) (2000) 379–391, http://dx.doi.org/10.1016/S0167-739X(99)00062-X.
[24] B. Raghavan, T. Kohno, A. Snoeren, D. Wetherall, Enlisting ISPs to improve online privacy: IP address mixing by default, Priv. Enhanc. Tech. (2009) 143–163, http://dx.doi.org/10.1007/978-3-642-03168-7_9.
[25] D. Raychaudhuri, K. Nagaraja, A. Venkataramani, Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet, ACM (2012) 2–13.
[26] M. Reed, P. Syverson, D. Goldschlag, Anonymous connections and onion routing, IEEE J. Sel. Areas Commun. 16 (4) (1998) 482–494, http://dx.doi.org/10.1109/49.668972.
[27] M. Reiter, A. Rubin, Crowds: Anonymity for web transactions, ACM Trans. Inf. Syst. Secur. 1 (1) (1998) 66–92, http://dx.doi.org/10.1145/290163.290168.
[28] Rohan, Cybersecurity market worth 231.94 billion USD by 2022, 2018, URL https://www.marketandmarkets.com/PressReleases/cyber-security.asp.
[29] Q. Wang, B. Qin, J. Hu, F. Xiao, Preserving transaction privacy in bitcoin, Future Gener. Comput. Syst. (2017) http://dx.doi.org/10.1016/j.future.2017.08.026.
[30] Y. Yang, X. Zheng, W. Guo, X. Liu, V. Chang, Privacy-preserving fusion of iot and big data for e-health, Future Gener. Comput. Syst. 86 (2018) 1437–1455, http://dx.doi.org/10.1016/j.future.2018.01.003.
[31] L. Zhang, Z. Yan, R. Kantola, Privacy-preserving trust management for unwanted traffic control, Future Gener. Comput. Syst. 72 (2016) 305–318, http://dx.doi.org/10.1016/j.future.2016.06.036.
[32] L. Zhang, Z. Yan, R. Kantola, Privacy-preserving trust management for unwanted traffic control, Future Gener. Comput. Syst. 72 (2017) 305–318, http://dx.doi.org/10.1016/j.future.2016.06.036.
[33] Y. Zhang, L.T. Yang, J. Chen, Security in wireless sensor networks, rfid and sensor networks: architectures, protocols, security, and integrations, 2009, pp. 415–460, http://dx.doi.org/10.1201/9781420077780.



Zhongjiang Yao is a Ph.D. student majoring in computer system architecture at the Institute of Information Engineering, Chinese Academy of Sciences. His current research interests include network security, blockchain and machine learning.

Jingguo Ge received a Ph.D. degree in computer system architecture from the Institute of Computing Technology of the Chinese Academy of Sciences in 2003. He is currently a professor at Institute of Information Engineering, Chinese Academy of Sciences, and a professor at School of Cyber Security, University of Chinese Academy of Sciences. His research focuses on computer network architecture, software defined network (SDN), network virtualization, cyber security, and mobile communication networks.



Yulei Wu is a Senior Lecturer in the Department of Computer Science with the University of Exeter, United Kingdom. He received the Ph.D. degree in Computing and Mathematics and the B.Sc. degree (Hons.) in Computer Science from the University of Bradford, United Kingdom, in 2010 and 2006, respectively. His expertise is on networking and his main research interests include intelligent networking technologies, network slicing and softwarization, future Internet architecture and technologies, Green networking, wireless networks, network security and privacy, and analytical modelling and performance optimization. He is an Editor of IEEE Transactions on Network and Service Management, Elsevier Computer Networks and IEEE Access. He contributes to major conferences on networking as various roles including a Steering Committee Chair, a General Chair, a Program Chair, and a Technical Program Committee Member. His research has been supported by Engineering and Physical Sciences Research Council of United Kingdom, National Natural Science Foundation of China, University's Innovation Platform and industries. He is a Senior Member of the IEEE, and a Fellow of the HEA (Higher Education Academy).

Linjie Jian received a master's degree in computer technology from the Institute of Information Engineering, Chinese Academy of Sciences in 2018. He is currently a research and development engineer at Baidu.com Inc.