EAI Endorsed Transactions
on Security and Safety

Research Article ICST.ORG

Towards Privacy-Preserving Web Metering Via User-Centric Hardware

Fahad Alarfi*,[1], Maribel Fernández[1]

[1]King's College London, Department of Informatics, Strand, London WC2R 2LS, UK.

Abstract

Privacy is a major issue today as more and more users are connecting and participating in the Internet. This paper discusses privacy issues associated with web metering schemes and explores the dilemma of convincing interested parties of the merits of web metering results with sufficient detail, and still preserving users' privacy. We analyse different categories of web metering schemes using an established privacy guideline and show how web metering can conflict with privacy. We propose a web metering scheme utilising user-centric hardware to provide web metering evidence in an enhanced privacy-preserving manner.

Received on 25 September 2014; accepted on 22 December 2014; published on 05 October 2015

Keywords: Web Metering, Privacy-Preserving Technologies, Privacy Protection, Cryptographic Hardware, Smart Cards And Privacy, Network & Distributed Systems Security

doi:10.4108/eai.5-10-2015.150478

1. Introduction

1.1. Web Metering Problem

Consider a service provider, which in the context of this paper will simply be a *webserver*, and a *user*, who is a person using a platform to access the webserver through an open network. The *web metering problem* is the problem of counting the number of visits done by such user to the webserver, additionally capturing data about these visits. Automated visits done by machines are outside the scope of this research partly because the research is mainly motivated by "Online Advertising". A *web metering scheme* produces the number of visits and supporting evidence to interested enquirers. The web metering scheme can be run by an *Audit Agency* or a less trusted third party *Metering Provider*. Figure 1 shows the four entities and their connections.

We classify web metering schemes as user-centric, webserver-centric or third-party-centric, depending on the entity controlling the scheme or having a major role in setting up the scheme.

Besides Online Advertising applications, the webserver might want to improve its content organisation to confidently allocate (or prioritise the display of) its content according to the web metering results. We consider a hostile environment where the adversary is motivated to fake users' visits or can invade users' privacy. The adversary can be a corrupt webserver or an outside attacker.

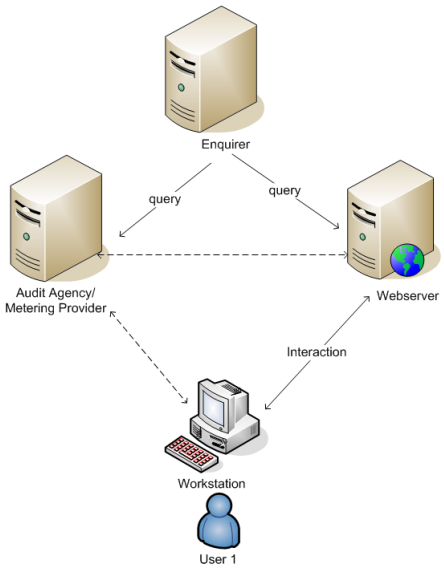

Figure 1. Web Metering Entities

Privacy is the right of individuals to control or influence what information related to them may be

*Corresponding author. Email: fahad.alarifi@kcl.ac.uk

collected and stored and by whom; and to whom that information may be disclosed [28]. Another stronger notion is unlinkability. Unlinkability of two or more items of interest requires that these items of interest are no more and no less related after the adversary observation [37]. In the web metering context, unlinkability of two or more user's visits requires that the observer cannot determine if the visits are related or not.

There are trade-offs between designing secure web metering schemes and preserving users' privacy. The schemes become more difficult to design when the main interacting party is not interested to participate and operations need to be carried out transparently. To satisfy such *transparency* property[1] [31], the scheme needs to execute inside or behind another existing action or property in the web interaction so it does not require a new explicit action from the user. Such user non-cooperation or simply disinterest further enforces low cost solutions that can provide web metering results without the user involvement or breach of his privacy rights. Also, determining the qualities of captured visits (e.g. time of each visit or age of the user) can be a requirement for some web metering applications and such granularity of data can help in disputes resolution regarding web metering evidence. However, it is a trade-off; the more non-repudiated information collected about the web interaction, the greater likelihood of invading users' privacy.

Following Dolev-Yao threat model [20], an adversary has also control over the communication channels and can obtain data sent through the channels, and send data to entities impersonating another entity. Consequently, an adversary could impersonate a valid user and could receive replies from the Audit Agency that contain private data about the user. The adversary could also capture (and possibly correlate) private data sent from the user. In addition, a corrupt webserver could store non-private data that could be correlated at a later stage and invade the users' privacy. Also, a corrupt webserver could ask for more information from the user and receive private data.

Despite the desired web metering granularity and the existence of adversary attacks, the concept of using privacy as an economic rationality [21] can be applied in the context of web metering. That is, web metering evidence can be generated by trading services to the user in exchange for information. This approach inherently has a limited scope because it assumes users wish to participate in the web metering scheme and therefore, it has questionable efficiency. However, when such benefits outweigh the risks, users tend to accept and adjust to metered interactions [24]; such an approach requires a web metering privacy policy for webservers to be able to fairly trade their services. Getting user information could be designed in new products so the activity happens transparently to the user. Furthermore, balancing the users' privacy right with the conflicting [9] webservers' and interested parties' freedom of expression right, complicates this interdisciplinary problem for privacy-preserving web metering schemes. Without closer analysis and specific metrics, service providers could pragmatically argue that information about visits to the webservers can be published as an exercise of their right of freedom of expression and for public interest.

[1] www.sites.google.com/site/yuriyarbitman/Home/on-metering-schemes

Paper Contribution. The contributions of this paper are as follows. We propose a new web metering scheme that uses a hardware device at the user side to provide web metering evidence in a privacy-preserving manner. To the best of our knowledge, the proposed scheme is the first generic hardware-based user-centric web metering scheme. We show that the proposed scheme has the required security properties and enhances the privacy of users. In addition, we show that, aside the presence of the hardware component, the scheme can be implemented in a way that makes web metering transparent to the user. We also use privacy measurements to analyse and compare different categories of web metering schemes, showing the benefits of the proposed scheme. This paper is an extended version of [1].

1.2. Related Work

User-centric schemes. User-centric web metering schemes can use *digital signatures* and hash chaining to construct non-repudiation evidences of visits as proposed by Harn and Lin [25]. To exempt the user from producing a costly signature for each visit, a hash chain is proposed. That is, the webserver uses the received signature and the hash values as evidence for the number of visits. However, the received signature can be linked to the user's identity, which is a privacy problem.

To avoid the apparent privacy problem with digital signatures, *Secret Sharing schemes* were proposed by Naor and Pinkas [35] and used in many works e.g. by Masucci [5, 6] and others [32, 42]. As evidence of the visits, the webserver here needs to receive a specific number of shares from users to be able to compute a required result using a Secret Sharing scheme e.g. Shamir Secret Sharing [40]. However, the user has to be authenticated (which is another privacy problem) so that the webserver cannot impersonate him and have the required shares. Also, if the Metering Provider is generating and sending the shares, it has

to be trusted not to collude with the webserver to link user identity with visits after the user-webserver interaction. Similarly, an adversary can observe and correlate user authentication data with the visits. The users' identities have also to be revealed to the Audit Agency to resolve disputes about collected shares by the webserver which can potentially be linked to the visits. With our assumptions, the proposed scheme addresses these issues.

Webserver-centric schemes. A webserver-centric *voucher* scheme uses e-coupons [29] as an attempt to map traditional advertisements models into the electronic ones. Such schemes can be used when a corrupt webserver is motivated to deflate number of visits [30]; however, the user has to be authenticated when forwarding the e-coupon to the issuing party to stop the webserver from forwarding the e-coupons itself. Also, a questionable Metering Provider can potentially use received e-coupons and authentication data and collude with the webserver to link the information to visits. Or an adversary can observe and correlate authentication data and e-coupons with the visits. Improvements have been proposed in [18] to address these issues in environments where the adversary is motivated to deflate number of visits. However, we only consider hit inflation attacks in this paper.

Another webserver-centric *processing-based* scheme was proposed by Chen and Mao [12] which uses computational complexity problems like prime factorisation. These computational problems attempt to force the webserver to use users resources in order to solve them and consequently provide web metering evidence via the produced result. However, besides using users' resources, an adversary can fake users' visits and possibly figure out computing resources at the user side e.g. by analysing the speed of the computations.

The use of a physical web metering *hardware* box attached to the webserver was proposed in [4]. In such webserver-centric scheme, the webserver connects to an audited hardware box which intercepts users requests and stores a log. Randomly, the box also produces a Message Authentication Code (MAC) on a user request which is then redirected to the Audit Agency as an additional verification step. The Audit Agency verifies the MAC code and the request and if valid, the received request is redirected back to the webserver. User impersonation is still a successful attack here in which the webserver can inflate the number of visits. The proposed scheme increases the cost of running a successful impersonation attack so that it is not feasible for a corrupt webserver.

Third-Party-Centric Schemes. A third-party-centric web metering scheme was proposed in [2] which tracks the user using an *HTTP proxy*. The intercepting HTTP proxy adds a JavaScript code to returned HTML pages to track users' actions e.g. mouse movements. Consequently, all visits have to go through the proxy, which does not preserve users' privacy.

Another third-party-centric scheme is *Google Analytics (GA)* [23] which can provide more granular information than the number of visits. During users' visits, referenced web metering code is loaded into the webserver script domain. The code is executed under the webserver control, setting a *webserver-owned* cookie [38] to track returning users to the webserver and not Google-Analytics.com. Then, the code extracts the user's assigned identifier in a cookie (set earlier or updated by the running script) and captures some user's data, all to be sent back to Google-Analytics.com. Despite the privacy improvement of webserver-owned cookie of not figuring out users visiting different webservers incorporating GA script, returning users will still be identified to the webserver and Google-Analytics.com. Besides the cookie issue, the referenced code captures private data about the user, e.g. user's Internet Protocol (IP) address to provide geographic results. In the proposed scheme, we ensure that such private information is preserved.

1.3. Paper Organisation

The remainder of this paper is organised as follows. Section 2 proposes a generic web metering scheme and provides an analysis covering assumptions, goals and practical aspects. Section 3 describes techniques to implement the generic scheme. Section 4 analyses the proposed scheme from security and privacy perspectives. Section 5 provides a proof-of-concept implementation analysis. Section 6 concludes the paper.

2. Web Metering Via User-Centric Hardware

2.1. High Level Description Of Proposed Scheme

Inspired by the webserver-centric hardware-based web metering scheme in [4] and the use of secure user-centric hardware-based broadcasting technique (e.g. pay television) in [14], we propose here a new web metering scheme that relies on a hardware device at the user side.

Definition 1. A **secure device** is an abstraction for an integrated circuit that can securely store a secret value. To access that secret value, a processor is needed which can be inside that hardware device or inside an attached computing platform. The device has to be equipped with a technique (e.g. *zeroization*) so that the secret key cannot be extracted.

The device contains a secured secret key used for authentication, and has the ability to store another signature secret key, inside or outside the device. Examples of such hardware devices are a smart card or an enhanced version e.g. a Trusted Platform Module (TPM) [27]. In addition to TPMs, two factor authentication is an authentication mechanism in which the user uses two different credentials e.g. a password and a hardware token. Banks two factor authentication token[2] is a non-transparent example of a hardware device distributed to the user for a secure webserver visit. The adversary could still *purchase* hardware devices for "fake" users' identities. The cost should typically be higher than the gained benefits, as in [22]. Our generic web metering scheme operates in an environment which consists of a webserver, a user, who owns a hardware device, and an Audit Agency. The three parties follow the protocol specified below. First, we define hardware authentication which will be used as a step in the generic scheme as follows.

Definition 2. Hardware authentication is a unilateral authentication [16] in which the Audit Agency is assured of the claimed communicating user's identity.

The following is a generic protocol for the proposed web metering scheme.

1. **User** → **Webserver** : Access request
2. **Webserver** → **User** : Certificate request
3. **User** → **Audit Agency** : Hardware certificate
4. **User** ↔ **Audit Agency** : Hardware authentication
5. **User** → **Audit Agency** : New key
6. **Audit Agency** → **User** : Certificate for new key
7. **User** ↔ **Webserver** : Certificate & signature
8. **Webserver** ↔ **Audit Agency** : Verification key & evidence

The protocol for the proposed web metering scheme consists of eight steps, as follows.

1. User sends an access request to webserver.
2. Webserver replies with a certificate request.
3. User sends the certificate for the secret key, inside the hardware device, to Audit Agency.
4. Audit Agency checks the received hardware device certificate. If the certificate is valid, Audit Agency authenticates the communicating user by checking whether he can access the corresponding hardware device.
5. If authentication succeeds, the user generates a signature key pair and sends public part of it to Audit Agency.
6. Audit Agency signs the received public key and sends a certificate back to user.
7. User signs webserver URL and time using the new signature key and sends the signed URL and certificate received in step 6 (or a form of it) to webserver.
8. Webserver checks the certificate and the signature, and stores them as evidence.

In step 1, the user sends an access request to an object in the webserver. In step 2, the webserver checks whether the user has submitted a valid (attestation) certificate. If not, the webserver requests a certificate signed by the Audit Agency. In step 3, the user checks if he holds a valid certificate. If so, step 7 is instead executed. Otherwise, the user sends to the Audit Agency, the certificate for the secret key embedded in the hardware device. For example, the *hardware certificate* can be a signature by a *hardware authority agency* (e.g. Intel) for a public key, where its corresponding private part is embedded in the hardware device. In step 4, the Audit Agency checks the validity of the received certificate (e.g. not revoked). If the certificate is valid, the Audit Agency checks whether the user holds the corresponding secret key in relation to the certificate. For this step, the user is asked to encrypt fresh nonces using the embedded secret key. In step 5, if the user is authenticated, he generates a new signature key pair and sends the public part of it (verification key) to the Audit Agency. This step can be executed for x number of key pairs. In step 6, the Audit Agency signs the received verification key (blindly if privacy is required) using its signature key and sends the produced signature (requested certificate) to the user. In step 7, the user forwards the received certificate in step 6 to the visited webserver or convinces the webserver that he has obtained a certificate. The user also sends his verification key to the webserver if it is not included in the submitted certificate. The user also signs a webserver identifier (e.g. URL) and possibly other information (e.g. time) and sends the signature to the webserver as evidence of the visit. For efficiency reasons, the webserver could periodically publish reference numbers or pseudonyms that can be linked to the webserver and time instead of concatenating URL and time for the *evidential signature*. In step 8, the webserver checks that the certificate was somehow *signed* using Audit Agency verification key. The webserver also checks (possibly using a privacy-preserving protocol) that the received signature was signed by the user's new signature key. If both

[2] www.hsbc.co.uk/1/2/customer-support/online-banking-security/secure-key

checks succeed, the webserver stores the certificate and signature as web metering evidence.

The following is an example of the proposed scheme. Assume some webservers can only be accessed if the users own web metering hardware devices. Once the user accesses a webserver at some point in time, the user gets redirected to an Audit Agency. The Audit Agency first ensures that the user owns a valid hardware device. Then, the user generates and sends a session key and asks for a certificate. The Audit Agency produces the certificate using different possible schemes and sends it to the user. The user gets redirected back to the webserver when he submits the certificate and a new signature message.

2.2. Security And Privacy Assumptions And Goals

We assume that number of corrupt users is small as done in [3]. In particular, the webserver cannot convince significant number of users to collude with it, to create fake web metering evidence. The rationale behind this assumption here is that the number of users captured by web metering evidence should typically be large and unlikely for the webserver to be able to cost-effectively motivate a considerable number of users into colluding. Also, colluding participants have to risk losing the functionality of their hardware devices once tagged as rogue.

User-centric hardware-based web metering schemes have a potential to overcome user impersonation attacks and can be designed to preserve users' privacy. This can be achieved by involving the Audit Agency in the user setup or increasing the cost of webserver faking visits, as followed in the lightweight security approach in [22]. The hardware introduction is used here to increase the cost for a corrupt webserver to fake visits by requiring it to own a hardware device for each fake user. At the same time, the scheme has to ensure that it is impossible for a corrupt webserver with one authentic hardware device to be able to generate an unlimited number of evidences e.g. using a periodic hardware authentication with a limit of issued certificates. Therefore, we need a hardware device at the user side containing a secret key. Also, the secret keys certificates and public cryptographic values have to be available to the Audit Agency as they are required in step 3. In steps 3 and 7, the user is assumed to be securely redirected and may not necessarily be aware of this ongoing web metering operation, if a privacy-preserving scheme is being used in a *transparent* mode.

A summary of the assumptions we followed in this paper are as follows.

1. Number of corrupt users is far less than the total number of metered users.
2. User owns a secure hardware device (as in Definition 1).
3. The Audit Agency can obtain a list of valid devices certificates (e.g. from Intel) and recognise revoked or expired ones. Alternatively, users could be incentivised to register their authentic hardware devices for privacy-preserving browsing.
4. The web metering environment is where the user's privacy is a concern.
5. There is limited value of the online content (affecting the cost for webserver owning hardware devices).

In the rest of this section we further describe attacks that can happen during a hostile web metering operation and then highlight the required security goals to counter such attacks. We derive the following security attacks from the adversary capabilities described in Dolev-Yao threat model [20]: replay, impersonation and man in the middle attacks.

Replay Attack. A replay attack occurs when an adversary captures data sent from the user to the Metering Provider, the Audit Agency or the webserver and sends the data again. Similarly, an adversary captures data sent from the webserver to the Metering Provider or the Audit Agency and sends the data again.

If a replay attack is not detected, the visits number may be increased.

Impersonation Attack. An adversary in an impersonation attack (which is more powerful than the replay attack scenario where attack effect is limited to captured data), creates fake data and sends it to the Metering Provider or the Audit Agency impersonating a valid webserver or user. Or an adversary creates a fake request to a webserver impersonating a valid user.

If an impersonation attack is not detected, the visits number may be increased or the evidence data may have invalid properties.

Man In The Middle Attack. Man in the middle attack occurs when an adversary receives data from the user or the webserver not intended to him and modifies it before forwarding it to the intended party.

If such attacks are not detected, the visits number may be increased or the data have invalid properties.

Besides the three communication attacks, there is also a threat that a corrupt webserver may not follow the required **web metering operations**. A corrupt webserver is inherently motivated to change the number of visits. Also, a corrupt webserver can be motivated to change some metering operations without changing number of visits. For example, a corrupt webserver intentionally changes a webpage identifier,

which is going to be recorded in the web metering evidence, to a different webpage that charges higher fees for advertisements.

To counter such attacks, there have to be data integrity of the web metering results and secure web metering operation (we define these concepts below). Data integrity is a property that counters threats to the validity of data [16]. Once this property is satisfied, it provides protection against unauthorised modification or destruction of data. Data Integrity in web metering refers to the integrity of stored and transferred evidences and data, as follows.

Evidential Integrity Goal. Evidential Integrity assures that evidences are kept as they were originally produced and stored. That is, once evidences are generated, they have to maintain their exact state and not change (maintaining evidences state includes intentional and accidental changes). Also, this integrity includes stored data that requires post processing work to produce the final web metering evidence.

Evidential integrity requirement is needed as a countermeasure to the web metering operation and stored web metering result attacks.

Communication Integrity Goal. Communication Integrity is concerned with integrity of the communication channels used for transferring web metering data. Transferred web metering data refers to pieces of data transmitted between users, webservers and Audit Agency that can be used to constitute the web metering evidence. Communicating this data has to be done in a way that if the data is changed en route, the change is going to be detected.

Communication integrity requirement is needed as a countermeasure to man in the middle communication channels attack.

The following security requirement is needed as a countermeasure to communication channels (replay and impersonation) and web metering operation attacks.

Security Goal. A web metering scheme is secure if its web metering operations are executed as expected per its specifications and can not be affected by an adversary, to eventually provide consistent results and evidence.

High Level Analysis Of Proposed Scheme. To ensure that an impersonation attack is not feasible, step 3 has been included as only users who have valid hardware devices will be authenticated (because the key cannot be extracted from the hardware device). As a result, an impersonation attack for imaginary set of users will require an adversary to own a hardware device for each impersonated user, which is not feasible. To ensure that man in the middle attack is not successful, step 4 has been included as an adversary listening to communications will not be able to satisfy the required authentication. Similarly, an adversary interfering the certificate in step 6 will not posses the corresponding secret part. To ensure that replay attack is not successful, time should be included in the signed messages. We provide a more detailed analysis of security properties of the scheme in section 4.1.

To preserve user's privacy, in step 6, the Audit Agency has to blindly sign the new user's key and send the blind signature (i.e. certificate) to the user. Owing to the blind signature production, the Audit Agency does not know the user's key. In step 7, the user submits a form of the received signature or proves to the webserver that she possesses an Audit Agency signature on the new web metering signature key. The webserver would store the signatures as evidence for number of visits that are done by users carrying authentic hardware devices. We provide a more detailed analysis of privacy properties of the scheme in section 4.2.

2.3. Practical Aspects

The use of hardware devices is common today. Commercial hardware tokens[3] can be used in the proposed scheme as long as they hold a *zeroizable* secret for authentication. Also, a relevant application that uses hardware decoders but not for web metering purposes, is pay television. Here, the user has to have hardware decoders to get multimedia content sent by a broadcasting server. Only authorised users' decoders can decrypt the broadcast content, using the embedded decryption keys. The server encrypts the broadcast content, which will be decrypted using the corresponding decryption key, inside the hardware decoder. The technique can also have other security properties like a tracing capability to detect rogue decoders that share the decryption keys [14].

In case the user is not motivated to explicitly participate in the web metering scheme but still have an applicable hardware device, the scheme can still be run transparently to the user, where a program (or a script[4]) anonymously attests the user. One current application requiring TPMs are digital wallets [13]. In a typical application, the user uses a virtual wallet program on a device to make a transaction. Potential motivations for such a wallet over credit cards could be finding better deals or further authenticating communicating users with customised information set in the wallet. Recent commercial devices (e.g. iPhone 6[5]) use Near-Field Communication (NFC) technology

[3] www.safenet-inc.com/uploadedfiles/about_safenet/resource_library/resource_items/product_briefs_-_edp/safenet_product_brief_ikey_4000.pdf
[4] www.cometway.com
[5] www.apple.com/iphone-6/

for such digital wallet. NFC devices can use TPMs [26]. On the other hand, an organisation might want to restrict accesses to their local network once users have certain hardware devices in a fashion similar to Virtual Private Network (VPN) connections. For example, the distributed hardware devices can provide the required connectivity and privacy-preserving web metering results. On a larger scale another non-transparent scenario could be to distribute free zeroizable devices to users (e.g. USB storage sticks).

There is also a trend of developing extra hardware devices (rather than traditional Personal Computers or mobile phones) for various desirable functions e.g. Google Glass[6]. Along the main functions like cameras or games, accessing certain webservers can be an additional function using a privacy-preserving web metering scheme.

3. Techniques To Implement Proposed Scheme

In this section, we start by describing mechanisms to implement each step in the proposed generic scheme.

Steps 1 and 2 can be implemented using standard mechanisms for issuing requests e.g. HTTP requests.

Steps 3 and 4 address the identification and authentication of the hardware device. As mentioned in section 2.1, a TPM can be used as a web metering hardware device for the required hardware authentication step. A *trusted computing* platform is a device which has an embedded TPM, which has Endorsement Key (EK) and a certificate on the public part of it to prove the platform is genuine. We can follow with such hardware device the lightweight security approach, where it is still possible for an adversary to construct fake web metering evidence but its cost does not offset the earned benefit.

Steps 5, 6 and 7 are included in the proposed scheme to take into account the privacy requirements. Step 8 is optional depending on whether the webserver needs to contact the Audit Agency for certificates or evidence redemption.

In the rest of this section, we describe existing protocols and schemes that can be used to implement steps 5, 6 and 7 in the web metering scheme in section 2.1. Using them, we give a technique to implement the scheme, satisfying both the security and users' privacy requirements. Table 1 outlines the mechanisms, satisfying the different requirements.

Security Without Privacy. To implement step 5, the following secure but *non-privacy-preserving* key transport protocol [7] can be used, where the Audit Agency and the user have public key pairs.

[6] www.google.com/glass/

1. **User** → **Audit Agency** : $ENC_{AuditAgency}[identifier, key, count]$
2. **Audit Agency** → **User** : $ENC_{key}[count,nonce]$
3. **User** → **Audit Agency** : $SIG_{User}[AuditAgency, H(nonce,count,key)]$

In the first message, the user encrypts, using a standard encryption scheme, his identifier, a new signature key and its count number using Audit Agency public key and sends it to the Audit Agency. The Audit Agency decrypts the message and encrypts the received count number and a nonce using the new key and sends it to the user. The user decrypts the message to reveal the nonce to hash it with the count number and the new key. Then, the user signs, using a standard signature scheme, the hash code along the Audit Agency identifier with his public key and sends the signature to the Audit Agency. This signed message can be used to link the new signature key to the user's identity. For step 6, once the new signature key is securely sent to Audit Agency, the Audit Agency signs it with its private key and sends the signature to the user as a certificate. For step 7, the user produces evidential signatures using the new signature key and uses the Audit Agency certificate to confirm its validity.

Security And Privacy. By contrast, to provide a privacy-preserving web metering scheme, the user has to commit to a new signature value for step 5 in the generic scheme, for example using Pedersen commitment scheme [36]. For the next step, an Audit Agency has to blindly sign the committed value (once the user is authenticated) and allow the user to prove its possession, without revealing it. For step 7, the user uses the new signature value, without linking it to the former authenticated credential.

A general view of the privacy-preserving technique required in step 5 can be two interacting entities in which one can prove to the other that it holds a secret without revealing it. New secrets can be generated with the help of a trusted third party while the former secret is "buried away" in another value. For example, using *Schnorr* zero-knowledge protocol [39], a secret s can be embedded in a smart card and used for signing such that $y = g^s mod\ p$ where g is a group generator and p and q are two large prime numbers such that q is a divisor of $p - 1$ (y, g, p and q are public values). A commitment scheme can be used in constructing a zero-knowledge protocol. In the web metering context, the user can convince the Audit Agency that the interacted messages are correctly formed using zero-knowledge proof of knowledge of a discrete logarithm. The following are the corresponding steps for Schnorr protocol, that can be used for step 5.

Table 1. Mechanisms Comparison

Security Without Privacy	Security And Privacy
Key transport protocol	Zero-knowledge protocol
Audit Agency signature	Audit Agency blind signature
User non-repudiation signature	User evidential signature of knowledge

1. **User** → **Audit Agency** : User chooses a random r and sends $a = g^r mod$ p
2. **Audit Agency** → **User** : Audit Agency sends a challenge c
3. **User** → **Audit Agency** : User sends $b = r + c * s$ mod q

In step 1, the user chooses a random value r where $1 \leq r \leq q-1$ and sends the commitment $a = g^r mod$ p. In step 2, the Audit Agency sends a challenge c where $1 \leq c \leq 2^t$ for some security parameter t. In step 3, the user sends to Audit Agency $b = r + c * s$ mod q. The Audit Agency checks whether $a * y^c = g^b$. If the check is correct, the Audit Agency is convinced that the user knows the secret s. The result of this check can be used as a proof that the user knows s without revealing it. For implementing step 6 in the generic scheme, the Audit Agency has to document the result as a "redeemable" privacy-preserving certificate. Then, for step 7, the zero-knowledge protocol has to run again between the user and the webserver.

In the following sections, we first show a secure technique that can be used to implement the generic scheme (without preserving the users' privacy). We then show a technique that can be used to implement the generic scheme, satisfying both the security and users' privacy requirements.

3.1. Privacy Certification Authority

In this part, we show a secure technique that can be used to implement the generic scheme (without preserving the users' privacy). The first attestation method published by Trusted Computing Group (TCG)[7] was to use Privacy Certification Authority (CA). Privacy CA has the same role as the Audit Agency. The following are seven steps using this attestation method for web metering purposes. In step 1, the user sends an access request to the webserver. In step 2, the webserver sends a request for attestation to the user, to enable the webserver to reply to the user request and reliably record web metering evidence. In step 3, the user submits his hardware certificate (for EK) to Privacy CA when the Privacy CA cross checks it with published certificates. All hardware certificates are initially published by the hardware authority agency and their status can be updated by the Privacy CA. In step 4, Privacy CA validates the used EK with respect to the submitted hardware certificate. In step 5, if the checks are correct, the user generates Attestation Identity Key (AIK) and sends the public part of it to the Privacy CA. In step 6, Privacy CA signs the received AIK and sends a signed AIK certificate. In step 7, the user uses AIK private key for producing evidential signatures to the webserver and the received AIK certificate to authenticate himself. Figure 2 shows the message flow for the described steps.

The privacy problem with using Privacy CA method is as follows. The webserver would send to Privacy CA (for example, on conflict resolution) the received signatures along the corresponding AIK certificate. Privacy CA can link the self-issued AIK certificate along the non-repudiation signatures to the corresponding used TPM's EK.

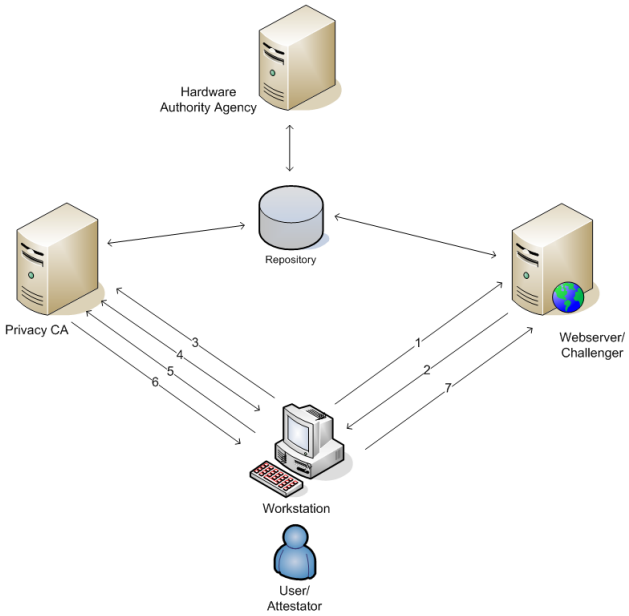

Figure 2. Attestation Using Privacy Certification Authority (CA)

3.2. Direct Anonymous Attestation Protocol

Direct Anonymous Attestation (DAA) protocol [8] can fortunately provide the needed public commitment, signature scheme and zero-knowledge proofs techniques. DAA protocol uses Camenisch-Lysyanskaya signature scheme [10] to provide a blind signature on

[7] www.trustedcomputinggroup.org

the committed value and allow the user to prove its possession, through a zero-knowledge proof of knowledge of the committed value. According to DAA protocol described in [8], communication between user and Audit Agency can be done using *Join Protocol* and communication between user and webserver can be done using *Sign/Verify Protocol*. Figure 3 shows the message flow for both protocols. Step *a* corresponds to steps 3 and 4 in the generic scheme. The rest of steps refer to the same order.

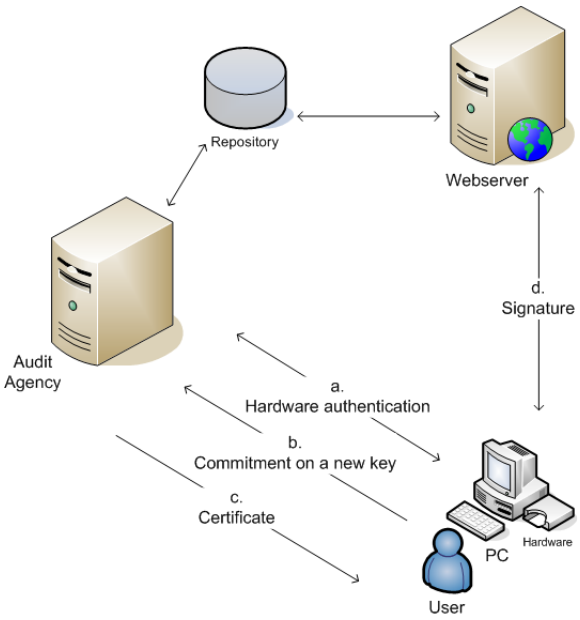

Figure 3. Anonymous Attestation Via User Hardware

The user gets authenticated to Audit Agency using EK (steps 3 and 4 in the generic scheme) and then receives a certificate as follows. In step 5, during Join Protocol, the user generates a secret key f and computes $U = z^f x^{v1} mod\ n$ where $v1$ is used to blind f and (n, x, y, z) is public key of Audit Agency. (z can be set-up as $x^{r2} mod\ n$ where $r2$ is random number so that the Audit Agency chosen random number will be multiplied by the secret f and added to the blind $v1$). Also, the user computes $N = Z^f\ mod\ p$ where Z is derived from Audit Agency identifier and p is a large prime. Then, the user sends (U, N) to the Audit Agency and convinces the Audit Agency that they are correctly formed using a proof knowledge of a discrete logarithm. We assume that the challenges and messages are securely chosen and constructed as specified in [8]. Then, in step 6 in the generic scheme, the Audit Agency computes $S = (y/(Ux^{v2}))^{1/e} mod\ n$ where $v2$ is random number and e is a random prime. Then, the Audit Agency sends $(S, e, v2)$ to the user to have (S, e, v) as a TPM certificate where $v = v1 + v2$. More than one secret can be generated here to guarantee unlinkability in case the Audit Agency is offline. The join phase is the heavy work phase of the scheme and can be periodically done for different requirements.

In step 7 in the generic scheme, during Sign/Verify Protocol, the user signs messages using the secret key f and Audit Agency certificate (S, e, v) received in Join Protocol. The user also computes $N2 = Z_2^f\ mod\ p$ where Z_2 is a group generator that can be configured for a required anonymity level. Z_2 can be fixed for a limited period of time in synchronisation with Audit Agency certificate issuance to determine unique number of users. For example, to determine unique users for a period of one hour, the Audit Agency has to keep a record of hardware authentications so the user cannot generate another key f, and Z_2 has to be fixed, for that period of time. Also, Z_2 can be chosen by the webserver, reflecting its identity e.g. a code for its URL. The b bit can be specified in DAA protocol to indicate that the signed message was chosen by the user. Such feature makes DAA more flexible to different desirable web metering applications than ecash [11]. Furthermore, a signed hash chain as in [25] can be used to efficiently know the length of the visited session and set it to a desired length. Consequently, the user can use the hash chain result as a generator. If the user is still online after the session ends, a new signed message (of a new hash chain) is created. This new message cannot be linked to the previous one as the webserver is just convinced that these messages were signed by user secret keys generated during the Join Protocol without the need to know them (proof of knowledge). The different generators capturing webserver URL and time with different certificates can guarantee (and tune) such required unlinkability and session length.

The user can provide a proof that she has a certificate for the secret values (f and v) by providing a zero-knowledge proof of the secret values, such that the following equation holds: $S^e z^f x^v \equiv y\ mod\ n$. Then, the user sends the signature to the webserver and convinces the webserver that she knows f, S, e and v. The webserver checks the signature and if valid, the webserver stores $N2$ along the result of the zero-knowledge proof as web metering evidence, proving interactively the communicated user's TPM was genuine.

4. Security And Privacy Analysis Of Proposed Scheme

In this section, we analyse and compare web metering schemes, starting with the proposed one in this paper.

4.1. Security Analysis Of Proposed Scheme

Proposition 1. An adversary capturing all communicated messages, but not owning the device, cannot:

1. create fake web metering evidence (i.e. N2, see section 3.2);
2. cannot impersonate an existing user.

Therefore, the proposed scheme achieves integrity and security goals.

Proof. We assume that the user owns a secure hardware device and number of corrupt users is small (as in section 2.2). Thus, hardware authentication (as in Definition 2) can only succeed by interactively proving the ownership of the physical hardware device containing the built-in secret key. Without a successful hardware authentication, valid evidence cannot be created in the absence of the subsequent committed signature key in step 5 (i.e. f). Consequently, the adversary has to own a hardware device in order to create valid web metering evidence. Moreover, we assume that the challenges and messages in steps 5, 6, and 7 are securely chosen and constructed as specified in section 3. Therefore, evidential integrity goal is achieved.

Depending on the Audit Agency setup, x certificates can be issued to the user after the successful hardware authentication, and valid for a limited period and cannot be reused. We assume that user's secret keys are used to encrypt nonces or time stamps, as specified in section 3, to ensure freshness as a countermeasure against impersonation and replay attacks for an observed user. While producing the zero-knowledge proof in proposed DAA-based scheme, the user has to interactively convince the other entity (Audit Agency or webserver) of the knowledge of secret values, using freshly provided challenges. Any captured messages that are resent again during Join Protocol will be rejected by Audit Agency as they will not fit in the current window of acceptable responses. Similarly, captured and resent messages during Sign/Verify Protocol will not enable webserver to construct new valid evidence N2 as they will not fit in the required window. Therefore, security goal is achieved.

Using zero-knowledge proof of a discrete logarithm [39], the adversary will not be able to learn the built-in secret key to pass the required authentication in Join Protocol nor be able to learn the corresponding secret signature key in Sign/Verify Protocol. Therefore, observing messages sent by a user will not enable the adversary to get the secret values to impersonate a valid user or hijack the session. Therefore, communication integrity goal is achieved.

4.2. Privacy Analysis Of Proposed Scheme

Proposition 2. The proposed DAA-based web metering scheme protects any identifying information captured from the authentic certificate of the user's hardware secret key.

Proof. By Definition 2, after hardware authentication, the Audit Agency is assured that the communicating user can securely access the secret key inside the hardware device and consequently can confirm the user's identity. Then, the zero-knowledge protocol [39] is used to convince the Audit Agency that constructed commitment messages were formed correctly without disclosing the secret value f. Once the Audit Agency is convinced, the Audit Agency produces a certificate S for the user's committed secret value f which is later anonymously used during Sign/Verify Protocol. Therefore, the proposed scheme protects any captured user's identifying information.

We assume during Sign/Verify phase, the user keeps the Audit Agency certificate (S, e, v) secret and only uses it to convince the webserver of the knowledge of the chosen secret key f. Otherwise, the Audit Agency can match the hardware certificate identifier to user's visits as follows. The Audit Agency records all issued certificates for the received hardware certificates during the "blind" signature production. Then, once the webserver has the exact Audit Agency certificates along users' signatures, the webserver forwards them to Audit Agency. The Audit Agency can check and match the self-issued certificates to the recorded hardware certificates.

During the Join Protocol, the user computes and sends U which is "embedded" in the Audit Agency certificate S. The user then convinces the webserver the knowledge of S along other secret values without disclosing S. The user also computes and sends N as a provision for recording and possibly revoking the approved commitments, as proposed in [8]. If such *linkability* feature is not required and lifetime of all approved fs is designed to be limited, user's computation and submission of N can be skipped. Similarly, there has to be a non-predictable difference in time or no pattern between user committing to a new signature key and using it. This is initially achieved by the two roles of Audit Agency and webserver when their involvement is separated by time. For example, when the user machine boots up, new keys are generated, approved and stored. For the next immediate visit, the user can either use previously approved signature keys or have to wait a random time before using the new keys. Then, the user is always set to contact the Audit Agency for new signature keys once the number of user's available keys goes below a threshold, say two. The random delay should be minimal as not to affect the user browsing experience. Also, to limit the effect of an impersonation attack, we can assume the scheme needs to re-run daily or every hardware boot up to limit the number of fake evidences.

The proposed scheme does not stop an adversary from capturing other non-required private information

about the user (e.g. IP address). A solution for such problem would be to use relevant security countermeasures (e.g. Network Addressing Translation [41] and Onion Network [19]) to prevent the capture of unrelated private data. In the rest of this section, we describe Network Address Translation and Onion Network.

Network Address Translation.

Using Network Address Translation (NAT), a unique address is changed to a different global one when communicated to another network. A NAT operation can be simplified as follows. A user inside a network makes a request to a webserver over the Internet. The user's request is first sent over the local network to the gateway (e.g. a router). The router changes the request machine IP address and source port into the global IP address and a new source port respectively. The router also records the request machine IP address and source port along the new assigned port. Then, the router does other required checks (e.g. integrity) before sending the user's request to the destination IP address and port specified in the request. When the router receives a reply, the router searches its record for the reply destination port address. If there is a match, the router extracts the corresponding user's IP address and port. Then, the router forwards the received reply to the extracted address and port.

Onion Network.

An *Onion Network* [19] is an alternative privacy-enhancing solution to proxies and home NATing devices especially in case the proxy is not trusted or not directly connected to the user (and consequently an adversary can observe the proxy's received requests). In such networks, the route from the user to the webserver is randomly set where each node en route only knows its predecessor and successor. The following are the protocol steps.

1. **User** → **Node 1** : $ENC_{pubkey}\ [x1 = g^{s1}]$
2. **Node 1** → **User** : $y1 = g^{s2}$, $HASH\ [K1 = g^{s1*s2}]$
3. **User** → **Node 1** → **Node 2**: $ENC_{K1}\ [x2 = g^{s3}]$
4. **Node 2** ⇒ **Node 1** → **User**: $ENC_{K1}\ [y2 = g^{s4}]$, $HASH\ [K2 = g^{s3*s4}]$
5. **User** → **Node 1** → **...** → **Node x**: $ENC_{K1,K2,,,Kx}$ [Connect To Webserver *identifier*]
6. **Node x** → **Webserver** : *TCP handshake*
7. **Node x** → **...** → **Node 1** → **User**: ENC_{Kx} [Successful Connection]
8. **User** → **Node 1** → **...** → **Node x**: $ENC_{K1,K2,,,Kx}$ [Get Webserver Object]
9. **Node x** → **Webserver** : Get Webserver Object
10. **Node x** → **...** → **Node 1** → **User**: ENC_{Kx} [Requested Webserver Object]

The user creates a secret key and negotiates Diffie Helmann parameters [17] with the first random node (Node 1). In step 1, the user sends $x1 = g^{s1}$ to Node 1 encrypted using Node 1 public key. In step 2, Node 1 sends back g^{s2} and a hash of the agreed Diffie Helmann key (g^{s1*s2}). Furthermore, from the agreed key, a key can be derived for each direction. In step 3, the user sends a request to Node 1 to extend the connection to Node 2. The request contains $x2$ for a new secret exponent, encrypted using the symmetric key $K1$. Once Node 1 receives the request, it decrypts it and forwards $x2$ to the next random node (Node 2). In step 4, Node 2 generates a secret ($s4$) and sends $y2$ and a hash of the agreed key ($K2$) to Node 1. Node 1 encrypts the received response using $K1$ and sends the encrypted message to the user. Then, the user calculates the new agreed key ($K2$) and checks the hash. Keys sharing and their forwarding in step 3 and 4 are repeated for further nodes e.g. Node 3. In step 5, the user sends a connect request to Node 1 encrypted using all agreed symmetric keys. In step 6, the end node (Node x) negotiates *TCP handshake* with the intended webserver, without encryption. In step 7, Node x sends a successful connection status message to the previous node in the path i.e. Node x-1, encrypted using the user and Node x agreed symmetric key. Similarly, the status message gets further encrypted down the path to user. Last, the user decrypts the received message using all agreed symmetric keys. In step 8, similar to step 6, the user sends webserver access request to Node 1 encrypted using all agreed symmetric keys. In step 9, Node x sends the received request to the webserver. In step 10, similar to step 7, Node x sends to the user the received webserver reply encrypted using the agreed key. Last, the user decrypts the received reply using using all agreed symmetric keys to reveal the requested webserver object.

The latest version of TOR (The Onion Router) browser should be used with its recommended settings. For example, Java[8] should be disabled so that Java circumventing attacks [34] are not successful. Otherwise an adversary's Java applet could surpass the onion network (or proxy) setup by making a direct connection to the webserver.

The following are two testing User Agents captured by a webserver for UK-based users. The locations of end nodes showed users' locations are instead Amsterdam (Netherlands) and Fremont (California) respectively. The operating system is the generic Windows NT 6.1 instead of MAC and Windows 7. Also, the browser was Mozilla instead of *Safari* and *Internet Explorer*.

```
% 31.5...141 is the user's IP address
```

[8] www.java.com

```
from a location in UK using Safari
browser on iPad to access the webserver.
The following is the user's request
to get the webserver homepage.

31.5...141 - - [22/Apr/2014:12:40:33
-0400] "GET / HTTP/1.1" 200 1897 "-"
"Mozilla/5.0 (iPad; CPU OS 7_0_4 like
Mac OS X) AppleWebKit/537.51.1 (KHTML,
like Gecko) Version/7.0 Mobile/11B554a
Safari/9537.53"

% The following is the user's request
to get the webserver homepage using
a TOR browser. 77.2...162 is the IP
address of the end node in Amsterdam.
The end node's browser is instead
Mozilla and operating system is
Windows NT 6.1.

77.2...162 - - [22/Apr/2014:12:41:17
-0400] "GET / HTTP/1.1" 200 1897 "-"
"Mozilla/5.0 (Windows NT 6.1; rv:24.0)
Gecko/20100101 Firefox/24.0"


% 137.7...8 is the user's IP address
from a location in UK using Internet
Explorer browser (MSIE 9) on Windows 7
operating system. The following is the
user's request to get the webserver
homepage.

137.7...8 - - [24/Apr/2014:10:21:57
-0400] "GET / HTTP/1.0" 200 1897
"-" "Mozilla/5.0 (compatible; MSIE 9.0;
Windows NT 6.1; Win64; x64; Trident/5.0)"

% The following is the user's request
to get the webserver homepage
using a TOR browser. 216.2...12 is the
IP address of the end node in Fremont.
The end node's browser is instead
Mozilla and operating system is
Windows NT 6.1.

216.2...12 - - [24/Apr/2014:10:39:11
-0400] "GET / HTTP/1.1" 200 1897
"-" "Mozilla/5.0 (Windows NT 6.1;
rv:24.0) Gecko/20100101 Firefox/24.0"
```

4.3. Privacy Analysis And Comparison Of Schemes

The World Wide Web Consortium (W3C) Platform for Privacy Preferences Project (P3P) [15] provides a framework regarding privacy issues in accessing webservers by allowing them to express their privacy practices in a standard format. We have analysed representative web metering schemes according to relevant metrics described in P3P. Further details about the compared schemes are provided in section 1.2. The following are the relevant P3P metrics and detailed schemes analysis.

- **Identifiers** are issued to users by a third party, which can be the Government or generally a "trusted" third party, to identify the user e.g. a username. There can be various levels of identifiers. For example, it is reasonable not to consider the action of capturing an IP address as identifying as authenticating an audited hardware decoder. Each user has a private and public key pair that is used to produce and verify the signature in the category of signature-based schemes. Also, GA assigns an identifier to the user browser after capturing user's IP address. Also, unencrypted traffic that goes through HTTP proxy including identifiers can be captured. In Secret Sharing schemes, secrets, submitted by the user, cannot be used as identifiers, however, users may have to be authenticated to get or verify the secrets. User hardware decoders techniques require users to have decryption keys upon subscription. Also, users, in webserver voucher schemes, have to be authenticated to ensure that the webservers are not forwarding the coupons themselves. Users' identifiers are required in the proposed scheme during hardware authentication in the Join Protocol. However, during the Sign/Verify Protocol, users' identifiers are preserved as previously used identification information, during Join Protocol, cannot be related thanks to the used zero-knowledge protocol.
- **State Management Mechanisms** are used to maintain the state of the connection to the webserver e.g. cookies. If the state of the user is required or can be captured, unlinkability cannot be provided by the scheme. In signature-based schemes, the user continuously submits a signature (or a hash value) that can link his visits. In typical Secret Sharing schemes, each user continuously submits a share to the webserver for each visit or session which link them up until the threshold value is computed. In user hardware decoders techniques, the state of the user can be tracked while decrypting on-the-fly broadcast content. Similarly, in webserver voucher schemes, the state of the user can be tracked as the user frequently forwards the e-coupons. The state of the user is continuously tracked in processing-based schemes because of

Table 2. Privacy Comparison

Scheme	**Identifiers**	State	Interactive	Location	Computer	Navigation
Digital Signature [25]	✗	↾	✓	✓	✓	✓
Secret Sharing [35]	✗	↾	✓	✓	✓	✓
Webserver Voucher [29]	✗	↾	✓	✓	✓	✓
Processing-Based [12]	✓	✗	✓	✓	↾	↾
Webserver Hardware [4]	✓	↾	✗	✓	✓	✓
HTTP Proxy [2]	↾	↾	↾	↾	✓	↾
Google Analytics [23]	↾	✗	✓	↾	↾	↾
This paper (DAA-Based [8])	✓	↾	✓	✓	✓	↾

the required participation from the user side. Also, user state can be figured out in HTTP Proxy schemes as all traffic goes through the proxy. Also, GA uses cookies to track the user state. The state of the user can be tracked in the proposed scheme depending on the key expiry date and the hash chain whenever used.

- **Interactive Data** includes data generated on the fly during user-webserver interaction e.g. a query to the webserver. User hardware decoders techniques can capture users' queries when encrypting particular responses (e.g. pay-per-view). Also, in webserver-centric hardware schemes, users' queries are occasionally captured and *MACed* before the result is forwarded to the Audit Agency. HTTP Proxy schemes can capture interactive data from the user as all traffic goes through the HTTP proxy.
- **Location Data** category covers information regarding the users location e.g. users' GPS location. Location of users is not directly captured in HTTP Proxy schemes, however, depending upon the location of the HTTP Proxy, location of users can be tracked. Also, GA captures IP address of the user which can have information about the user location. From the described user hardware decoders technique, we infer that users' location data cannot be captured.
- **Computer Information** is any information about the user computer e.g. IP address. Processing-based schemes and GA can capture computer information by analysing the time needed to return the solution (e.g. estimated CPU speed during the visit) and capturing the IP address respectively. Users' computer information can only be captured in the proposed scheme during the Join protocol by figuring out information regarding the platform from the hardware certificate.
- **Navigation and Click-stream Data** covers data about the user browsing behaviour e.g. user clickstream. Depending on the implementation of the processing-based scheme, navigation data can be required (e.g. user presence is determined by mouse movements). Also, HTTP proxy returns to the user a tracking code that captures the user mouse movement. Also, GA captures navigation data about the user through the type of referral which is stored in the cookie. Depending on the proposed scheme implementation, navigation data can be captured. For example, if the signed webserver URL references various levels within the webserver content, navigation data can be captured during the lifetime of the used key f.

A summary of the P3P analysis is shown in Table 2. From two extremes, a particular private information can be either *required* by the scheme or *protected*. We use the symbol ✗ to denote the scheme cannot operate without the corresponding required private information in order to provide web metering result or evidence. On the other hand, we use the symbol ✓ to denote that the private information can be protected and not accessed by the adversary under secure user setup. Such setup can be achieved with countermeasures that can prevent the adversary from getting the private information. The countermeasures can be provided by the scheme itself or can be potentially provided by other techniques. An example of outside countermeasures that can prevent the adversary from getting protected information could be a user behind a firewall with anonymous browser settings. We use the symbol ↾ to denote that the private information is not always or necessarily required by the web metering scheme; however, it is *available* and can still be captured by the adversary. Such available information can still be captured due to an efficient implementation (or a variation) of the scheme.

It can be observed from the analysis summary that the categories Identifiers and State are the least satisfied privacy categories. In particular, all schemes require or can capture the state of the user. Furthermore, once a user is identified or tracked, other private information e.g. user's preferences can be captured from the webserver content. If a scheme was able to capture the users' state but was not able to identify the user,

the captured state alone is not considered a privacy concern. Identifier information is used to achieve security requirements however such authentication information inherently conflicts with privacy. Identifier information can be the determinant metric to provide a privacy-preserving scheme. Processing and webserver hardware-based schemes are the most satisfying privacy-preserving schemes as they do not require nor can capture users' identifiers.

5. Proof-Of-Concept Implementation Analysis

We tested[9] the proposed scheme from the user side. We did various optimised simulation tests on a user machine with 2727578 high resolution performance counter frequency, using standard *math*, *gmp* and *OpenSSL* libraries[10] and the bit length specified in [8]. The prime numbers were of 1024 bits and modulus and generators were of 2048 bits. The public key values can be precomputed and stored at the hardware device, or securely downloaded to the user.

The tests showed feasible results. It took around 1650 nanoseconds to execute the first part of the public commitment (U) and around 515 nanoseconds to execute the second part (N). Then, the certificate production equation (S) needs only to be executed at the Audit Agency side, possibly by utilising Montgomery reduction algorithm [33] since the equation requires floating number exponentiation and *div* operations. From the results, the operations throughout the scheme phases can be executed in a short time so that they are not noticeable to the user.

The kind of secure hardware device required by the scheme is commonly used today. For example, the BitLocker program[11] uses the TPM public key for disk encryption, allowing the decryption (by TPM private key) if baseline platform measurements are met again. Furthermore, all Windows systems are planned to be shipped with TPMs in order to pass hardware certification of the latest operating system [43]. The scope of the required hardware devices is not limited to TPMs as discussed in section 2.3.

6. Conclusion

Privacy-preserving web metering is a challenging problem, especially with current necessary trade-offs and in environments where the Audit Agency could collude with webservers to identify users and link their visits. In this paper, we proposed an alternative and new user-centric web metering scheme using hardware to enhance users' privacy. We described a generic web metering scheme of a straightforward protocol and a special case using an existing protocol. We also used established privacy guidelines to analyse and compare representative categories of web metering schemes and showed the gained privacy benefits of the proposed scheme. The proposed scheme can provide different security countermeasures and users' privacy settings. However, denial of service attacks are still possible.

We built a proof of concept implementation on a traditional computer to evaluate the efficiency and transparency of the running web metering operations. Besides the operational cost from Audit Agency and webserver sides, the main barrier for a wide deployment is that users should accept the hardware device. However, in many contexts, the gain in privacy will offset the costs. We discussed how the user hardware assumption can be realistic in today's and future computing devices and showed different options.

Future work includes exploring techniques for discovering rogue hardware devices, and implementing the scheme with different settings to provide the evidential signature e.g. hash chaining. Various options for counting the number of unique users can be further explored for different advertising applications. Future work also includes analysing the performance of the proposed scheme using handheld devices. Formal validation of the proposed scheme is left for future work as well.

[9] at National Center for Digital Certification (NCDC): Research & Development. www.ncdc.gov.sa

[10] www.openssl.org/docs/

[11] www.technet.microsoft.com/en-us/library/cc162804.aspx

References

[1] F. Alarifi and M. Fernández. Towards privacy-preserving web metering via user-centric hardware. In *First Workshop on Secure Smart Systems (SSS 2014)*, September 2014.

[2] R. Atterer, M. Wnuk, and A. Schmidt. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 203–212, New York, NY, USA, 2006. ACM.

[3] S. G. Barwick, W.-A. Jackson, and K. M. Martin. A general approach to robust web metering. *Des. Codes Cryptography*, 36(1):5–27, 2005.

[4] F. Bergadano and P. D. Mauro. Third party certification of http service access statistics (position paper). In *Security Protocols Workshop*, pages 95–99, 1998.

[5] C. Blundo, A. D. Bonis, and B. Masucci. Bounds and constructions for metering schemes. In *Communications in Information and Systems 2002*, pages 1–28, 2002.

[6] C. Blundo, S. Martìn, B. Masucci, and C. Padrò. A linear algebraic approach to metering schemes. Cryptology ePrint Archive, Report 2001/087, 2001.

[7] C. Boyd and D.-G. Park. Public key protocols for wireless communications. In *Proceedings of The 1st International Conference on Information Security and Cryptology, ICSCI '98, Seoul, Korea*, pages 47–57, December 1998.

[8] E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, pages 132–145, New York, NY, USA, 2004. ACM.

[9] Calcutt, David. Committee on Privacy and Related Matters. *Report of the Committee on Privacy and Related Matters*. Cm (Series) (Great Britain. Parliament). H.M. Stationery Office, 1990.

[10] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '02, pages 61–76, London, UK, UK, 2002. Springer-Verlag.

[11] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Proceedings on Advances in Cryptology*, CRYPTO '88, pages 319–327, New York, NY, USA, 1990. Springer-Verlag New York, Inc.

[12] L. Chen and W. Mao. An auditable metering scheme for web advertisement applications. In *ISC*, volume 2200 of *Lecture Notes in Computer Science*, pages 475–485. Springer, 2001.

[13] Y. Chen, M. Moinuddin, and Y. Yacobi. Mobile wallet and digital payment, Sept. 30 2011. US Patent App. 13/249,381.

[14] B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Transactions on Information Theory*, 46(3):893–910, 2000.

[15] L. Cranor, B. Dobbs, S. Egelman, G. Hogben, J. Humphrey, M. Langheinrich, M. Marchiori, M. Presler-Marshall, J. Reagle, M. Schunter, D. A. Stampley, and R. Wenning. The platform for privacy preferences. W3C Recommendation, November 2006.

[16] A. W. Dent and C. J. Mitchell. *User's Guide To Cryptography And Standards (Artech House Computer Security)*. Artech House, Inc., Norwood, MA, USA, 2004.

[17] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[18] X. Ding. A hybrid method to detect deflation fraud in cost-per-action online advertising. In *Applied Cryptography and Network Security, 8th International Conference, ACNS 2010, Beijing, China, June 22-25, 2010. Proceedings*, pages 545–562, 2010.

[19] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, pages 21–21, Berkeley, CA, USA, 2004. USENIX Association.

[20] D. Dolev and A. C. Yao. On the security of public key protocols. Technical report, Stanford, CA, USA, 1981.

[21] P. Dourish and K. Anderson. Collective information practice: Exploring privacy and security as social and cultural phenomena. *Human Computer Interaction*, 21(3):319–342, 2006.

[22] M. K. Franklin and D. Malkhi. Auditable metering with lightweight security. *Journal of Computer Security*, 6(4):237–256, 1998.

[23] Google. Google analytics blog. Official weblog offering news, tips and resources related to google's web traffic analytics service. analytics.blogspot.com.

[24] J. Grudin. Desituating action: Digital representation of context. *Human Computer Interaction*, 16(2-4):269–286, 2001.

[25] H. Harn, L. Lin. A non-repudiation metering scheme. *Communications Letters, IEEE*, 37(5):486–487, 2001.

[26] M. Hutter and R. Toegl. A trusted platform module for near field communication. In *Proceedings of the 2010 Fifth International Conference on Systems and Networks Communications*, ICSNC '10, pages 136–141, Washington, DC, USA, 2010. IEEE Computer Society.

[27] International Organization for Standardization. ISO 11889-1:2009. *Information technology – Trusted Platform Module – Part 1: Overview*, May 2009.

[28] International Organization for Standardization. ISO 7498-2:1989. *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 2: Security Architecture*, 1989.

[29] M. Jakobsson, P. D. MacKenzie, and J. P. Stern. Secure and lightweight advertising on the web. *Computer Networks*, 31(11-16):1101–1109, 1999.

[30] R. Johnson and J. Staddon. Deflation-secure web metering. *International Journal of Information and Computer Security*, 1(1/2):39–63, 2007.

[31] R. Kumar. *Human Computer Interaction*. Laxmi Publications, 2005.

[32] C.-S. Laih, C.-J. Fu, and W.-C. Kuo. Design a secure and practical metering scheme. In *International Conference on Internet Computing*, pages 443–447, 2006.

[33] P. L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, 1985.

[34] J. A. Muir and P. C. V. Oorschot. Internet geolocation: Evasion and counterevasion. *ACM Comput. Surv.*, 42(1):4:1–4:23, Dec. 2009.

[35] M. Naor and B. Pinkas. Secure and efficient metering. In *EUROCRYPT*, pages 576–590, 1998.

[36] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '91, pages 129–140, London, UK, 1992. Springer-Verlag.

[37] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In *Workshop on Design Issues in Anonymity and Unobservability*, pages 1–9, 2000.

[38] F. Roesner, T. Kohno, and D. Wetherall. Detecting and defending against third-party tracking on the web. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 155–168, San Jose, CA, 2012. USENIX.

[39] C.-P. Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, pages 239–252, 1989.

[40] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[41] P. Srisuresh and K. Egevang. IP Network Address Translator (NAT) Terminology and Considerations, August 1999.

[42] R.-C. Wang, W.-S. Juang, and C.-L. Lei. A web metering scheme for fair advertisement transactions. *International Journal of Security and Its Applications*, 2(4):453–456, October 2008.

[43] Windows Certification Program. *Hardware Certification Taxonomy & Requirements*, April 2014.