Towards Privacy-Preserving Web Metering via User-Centric Hardware

Fahad Alarifi(✉) and Maribel Fernández

Department of Informatics, King's College London, Strand,
London WC2R 2LS, UK
{fahad.alarifi,maribel.fernandez}@kcl.ac.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 propose a web metering scheme utilising user-centric hardware to provide web metering evidence in an enhanced privacy-preserving manner.

Keywords: Web metering · Privacy · Secure embedded hardware

1 Introduction

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. A *web metering scheme* produces the number of visits and supporting evidence to interested enquirers, mainly for *Online Advertising* applications. The web metering scheme can be run by an *Audit Agency* or a less trusted third party *Metering Provider*. There are three different classes of web metering schemes, each with its own problems. Web metering schemes are classified 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. 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.

Privacy is the right of individuals to control or influence what information related to them may be collected and stored and by whom; and to whom that information may be disclosed [17]. 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, the scheme needs to execute inside or behind another existing action or property so it does not require a new explicit action from the user.

J. Tian et al. (Eds.): SecureComm 2014, Part II, LNICST 153, pp. 329–340, 2015.
DOI: 10.1007/978-3-319-23802-9_25

Contributions. 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.

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 [10], 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 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. In addition to the secured secret key, we assume that another signature secret key will be stored 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) [16]. The adversary could still *purchase* devices for "fake" users' identities. The cost should typically be higher than the gained benefits, as in [14].

Our generic web metering scheme operates in an environment which consists of a webserver, a user, who owns a 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.

Definition 2. *Hardware authentication is a unilateral authentication [12] in which the Audit Agency is assured of the 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

In step 1, the user sends an access request to the webserver. In step 2, the webserver checks whether the user has submitted a valid (attestation) certificate. If not, the webserver requests a certificate (to be issued from the Audit Agency). In step 3, the user checks if she 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 device. In step 4, the Audit Agency checks the validity of the received certificate (e.g. not revoked) and 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, the user 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 she has obtained a certificate. The user also sends her 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 *evidential signature* to the webserver. 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 checks succeed, the webserver stores the certificate and signature as web metering evidence.

2.2 Security and Privacy Assumptions and Attacks

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.

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 [14]. The use of hardware increases the cost for a corrupt webserver to fake visits by requiring it to own a 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 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 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 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 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 [13]: replay, impersonation and man in the middle attacks.

Attack 1. *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.

Attack 2. *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.

Attack 3. *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 web metering evidence, to a different webpage that charges higher fees for advertisements.

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 devices. In Sects. 4 and 5, we provide a more detailed analysis of security and privacy properties of the scheme and show that the attacks are not possible.

2.3 Practical Aspects

The use of hardware devices is common today. Commercial hardware tokens can be used in the proposed scheme as long as they hold a *zeroizable* secret for authentication. 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 [10].

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) anonymously attests the user. For example, the *BitLocker program* uses the TPM public key for disk encryption, allowing the decryption (by TPM private key) if baseline platform measurements are met again. Another current application requiring TPMs are *digital wallets*. 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. On the other hand, an organisation might want to restrict accesses to their local network once users have certain devices in a fashion similar to Virtual Private Network (VPN) connections. For example, distributed 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) which they could use for privacy-preserving web browsing. There is also a trend of developing hardware devices (rather than traditional Personal Computers or mobile phones) for various desirable functions e.g. *Google Glass*. 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 the 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 device. As mentioned in Sect. 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 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. In Sects. 3.1 and 3.2, we describe existing protocols and schemes that can be used to implement steps 5, 6 and 7 in the web metering scheme defined in Sect. 2.1. Using them, we obtain a technique to implement the scheme, satisfying both the security and users' privacy requirements. **Step 8** is optional depending on whether the webserver needs to contact the Audit Agency for certificates or evidence redemption.

3.1 Security and Privacy Techniques for Steps 5, 6 and 7

To provide a privacy-preserving web metering scheme, the user has to commit to a new key for step 5 in the generic scheme e.g. using Pedersen commitment scheme [21]. 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 [23], 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. We discuss in Sect. 3.2 a technique to implement step 6 in the generic scheme where 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.

3.2 Direct Anonymous Attestation Protocol for Steps 5, 6 and 7

Direct Anonymous Attestation (DAA) protocol [7] can fortunately provide the needed public commitment, signature scheme and zero-knowledge proofs techniques. DAA protocol uses Camenisch-Lysyanskaya signature scheme [8] to provide a blind signature on 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 [7], 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*.

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 [7]. 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 true identity. The b bit can be specified in DAA protocol to indicate that the signed message was chosen by the user.

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 Analysis of Proposed Scheme

We assume that the user owns a secure device and number of corrupt users is small (as in Sect. 2.2). Thus, hardware authentication (as in Definition 2) can only succeed by interactively proving the ownership of the physical device containing the built-in secret key. 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 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 Sects. 3.1 and 3.2. 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 Sects. 3.1 and 3.2, to ensure freshness as a countermeasure against impersonation and replay attacks for an observed user. 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 [23], 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. Consequently, the following proposition holds.

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

1. *create fake web metering evidence (i.e., N2, see Sect. 3.2);*
2. *impersonate an existing user.*

5 Privacy Analysis of Proposed Scheme

By Definition 1, after hardware authentication, the Audit Agency is assured that the communicating user can securely access the secret key inside the device and consequently can confirm the user's identity. Then, the zero-knowledge protocol [23] is used to convince the Audit Agency that constructed commitment messages were formed correctly without disclosing the secret value f. We assume that 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. 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. (Any introduced random delay should be minimal as not to affect the user browsing experience). Therefore, the proposed scheme protects any captured identifying information. With our assumptions, the following proposition holds.

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.*

6 Related Work

User-centric Web Metering Schemes. User-centric schemes can use *digital signatures* and hash chaining to construct non-repudiation evidences of visits as proposed by Harn and Lin [15]. 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 [20] and used in many works e.g. by Masucci [5,6] and others [19,25]. 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 [24]. 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. 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.

Webserver-centric Web Metering Schemes. A webserver-centric *voucher* scheme uses e-coupons [18] as an attempt to map traditional advertisements models into the electronic ones. 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. Another webserver-centric *processing-based* scheme was proposed by Chen and Mao [9] 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 still fake users' visits.

The use of a physical web metering *hardware* box attached to the webserver was proposed in [4]. 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.

Third-party-centric Web Metering Schemes. A third-party-centric 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 scheme is *Google Analytics (GA)* [1] which can provide more granular information than the number of visits. However, during the user-webserver interaction, GA captures private information about the user e.g. Internet Protocol (IP) address to provide geographic results. 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 [22] to track returning users to the webserver and not 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. Also, the referenced code captures private data about the user e.g. user's Internet Protocol (IP) address to provide geographic results.

Privacy Comparison. The World Wide Web Consortium (W3C) Platform for Privacy Preferences Project (P3P) [11] 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. A summary of the P3P analysis is shown in Table 1. From two extremes, a particular private information can be either *required* by the scheme or *protected.* We use the symbol ✗ to denote the

Table 1. Privacy comparison

Scheme	**Identifiers**	State	Interactive	Location	Computer	Navigation
Digital Signature [15]	✗	∤	✔	✔	✔	✔
Secret Sharing [20]	✗	∤	✔	✔	✔	✔
Webserver Voucher [18]	✗	∤	✔	✔	✔	✔
Processing [9]	✔	✗	✔	✔	∤	∤
Webserver Hardware [4]	✔	∤	✗	✔	✔	✔
HTTP Proxy [2]	∤	∤	∤	∤	✔	∤
Google Analytics [1]	∤	✗	✔	∤	∤	∤
This paper (DAA [7])	✔	∤	✔	✔	✔	∤

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. We use the symbol $\nmid$ 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 due to an implementation (or a variation) of the scheme.

7 Conclusion

We proposed a new user-centric web metering scheme using hardware to enhance users' privacy. We built a proof of concept implementation[1] on a traditional computer to evaluate efficiency and transparency of running operations. The tests showed feasible results. It took around 1650 nanoseconds to execute U and around 515 nanoseconds to execute N. Besides operational cost from Audit Agency and webserver sides, main barrier for a wide deployment is that users should accept the device. However, in many contexts, gain in privacy will offset the costs. We discussed how 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 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.

References

1. Google analytics blog. Official weblog offering news, tips and resources related to google's web traffic analytics service. analytics.blogspot.com
2. Atterer, R., Wnuk, M., Schmidt, A.: Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: WWW 2006: Proceedings of the 15th International Conference on World Wide Web, pp. 203–212. ACM, New York (2006)
3. Barwick, S.G., Jackson, W.-A., Martin, K.M.: A general approach to robust web metering. Des. Codes Crypt. **36**(1), 5–27 (2005)
4. Bergadano, F., De Mauro, P.: Third party certification of HTTP service access statistics (Position Paper). In: Christianson, B., Crispo, B., Harbison, W.S., Roe, M. (eds.) Security Protocols 1998. LNCS, vol. 1550, pp. 95–99. Springer, Heidelberg (1999)

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

5. Blundo, C., Bonis, A.D., Masucci, B.: Bounds and constructions for metering schemes. Commun. Inf. Syst. **2**, 1–28 (2002)
6. Blundo, C., Martn, S., Masucci, B., Padr, C.: A linear algebraic approach to metering schemes. Cryptology ePrint Archive, Report 2001/087 (2001)
7. Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, pp. 132–145. ACM, New York (2004)
8. Camenisch, J.L., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 61–76. Springer, Heidelberg (2002)
9. Chen, L., Mao, W.: An auditable metering scheme for web advertisement applications. In: Davida, G.I., Frankel, Y. (eds.) ISC 2001. LNCS, vol. 2200, pp. 475–485. Springer, Heidelberg (2001)
10. Chor, B., Fiat, A., Naor, M., Pinkas, B.: Tracing traitors. IEEE Trans. Inf. Theory **46**(3), 893–910 (2000)
11. Cranor, L., Dobbs, B., Egelman, S., Hogben, G., Humphrey, J., Langheinrich, M., Marchiori, M., Presler-Marshall, M., Reagle, J., Schunter, M., Stampley, D.A., Wenning, R.: The platform for privacy preferences. W3C Recommendation, November 2006
12. Dent, A.W., Mitchell, C.J.: User's Guide to Cryptography and Standards. Artech House Computer Security. Artech House Inc., Norwood (2004)
13. Dolev, D., Yao, A.C.: On the security of public key protocols. Technical report, Stanford, CA, USA (1981)
14. Franklin, M.K., Malkhi, D.: Auditable metering with lightweight security. J. Comput. Secur. **6**(4), 237–256 (1998)
15. Harn, H., Lin, L.: A non-repudiation metering scheme. IEEE Commun. Lett. **37**(5), 486–487 (2001)
16. International Organization for Standardization: ISO 11889–1:2009. Information technology - Trusted Platform Module - Part 1: Overview, May 2009
17. International Organization for Standardization: ISO 7498–2:1989. Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture (1989)
18. Jakobsson, M., MacKenzie, P.D., Stern, J.P.: Secure and lightweight advertising on the web. Comput. Netw. **31**(11–16), 1101–1109 (1999)
19. Laih, C.-S., Fu, C.-J., Kuo, W.-C.: Design a secure and practical metering scheme. In: International Conference on Internet Computing, pp. 443–447 (2006)
20. Naor, M., Pinkas, B.: Secure and efficient metering. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 576–590. Springer, Heidelberg (1998)
21. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
22. Roesner, F., Kohno, T., Wetherall, D.: Detecting and defending against third-party tracking on the web. In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, NSDI 2012, p. 12. USENIX Association, Berkeley (2012)
23. Schnorr, C.-P.: Efficient identification and signatures for smart cards. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 239–252. Springer, Heidelberg (1990)
24. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
25. Wang, R.-C., Juang, W.-S., Lei, C.-L.: A web metering scheme for fair advertisement transactions. Int. J. Secure. Appl. **2**(4), 453–456 (2008)