Direct Anonymous Attestation: Enhancing Cloud Service User Privacy

Ulrich Greveler, Benjamin Justus, and Dennis Loehr

Computer Security Lab
Münster University of Applied Sciences
D-48565 Steinfurt, Germany
{greveler,benjamin.justus,loehr}@fh-muenster.de

Abstract. We introduce a privacy enhancing cloud service architecture based on the Direct Anonymous Attestation (DAA) scheme. In order to protect user data, the architecture provides cloud users with the abilities of controlling the extent of data sharing among their service accounts. A user is then enabled to link Cloud Service applications in such a way, that his/her personal data are shared only among designated applications. The anonymity of the platform identity is preserved while the integrity of the hardware platform (represented by Trusted Computing configuration register values) is proven to the remote servers. Moreover, the cloud service provider can assess user account activities, which leads to efficient security enforcement measures.

Keywords: Trusted Management, Privacy Enhancing, Direct Anonymous Attestation, Cloud Services.

1 Introduction

1.1 Cloud Services

Cloud Services are becoming ever more popular because they offer users mobility, scalability and reliability. For instances, users are able to access computer systems using a web browser regardless of their locations or what devices they are using. And as the cloud infrastructure is off-site (typically offered by the third party), it drastically reduces the operational expenditure of a business.

Google is currently one of the most popular Cloud Service providers. The Google business model is likely to set a trend for future Cloud Service providers. Under a Google account, a user is able to access his registered services offered by Google. This includes the popular *Gmail*, *Picasa Web Album*, and *Google Talk*. Google offers other tools, we shall name a few more as they pertain to the privacy discussion in the next section. *Google Buzz* [2] is a social networking and messaging tool integrated with *Gmail*. *Google Latitude* [3] is a geo-location tool that allows friends to know where the user is via *Google Map* (Figure 1). The tool has a "Location History" feature which stores and analyzes a user's location (via user's mobile phone) over time, and it also attempts algorithmically to determine where a person lives and works and other information relating to a person's profile [23].

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011, Part II, LNCS 7045, pp. 575–585, 2011.

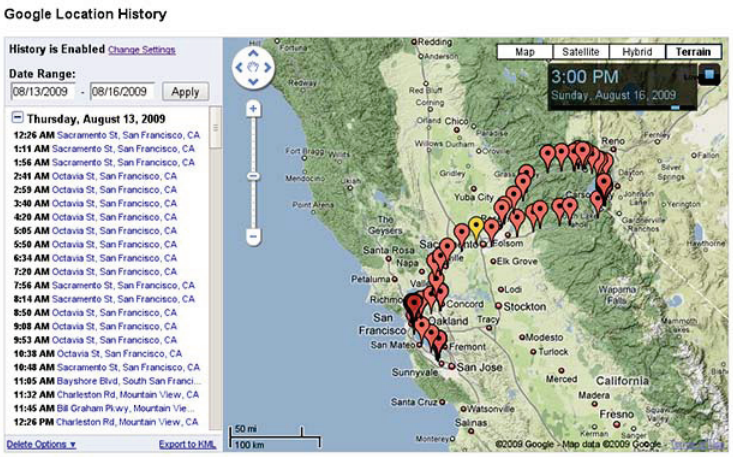

Fig. 1. Location History feature in Google Latitude

1.2 Privacy Concerns

Cloud Services often use a centralized approach when it comes to data storage of an individual's user account. For example, *Google Dashboard* [1] is a place where a user can login and view data that Google services have collected about the user. This includes user-provided information (names, addresses, profiles), and data contained in each of the service accounts (emails in *Gmail*, events in *Google Calendar*, photos in *Picasa*).

Some of the Google services have already drawn criticisms [22,7] because of privacy concerns. As Cloud Services such as Google collect more and more personal data and store them in a centralized manner, the consequence of exposing or leaking an account's information could be nightmarish. Just imagine someone taps into your account, can by clicking mouse a few times, discover who you are (name in the profile), where you live (home address), where you work (working address), who your friends are (mail contacts or *Google Buzz*), what your habits are (*Google Latitude*), and your financial data(*Google Checkout*). Such a scenario is not fiction-writing. According to Google [21] and its transparency report [4], the company receives constant requests from governments around the world to provide information on Google Service users. It is desirable that some measures of data control are available on the part of users.

1.3 Trusted Computing Background

The Trusted Computing Group (TCG) is an industry standards body formed to develop and promote specifications for trusted computing and security technologies. Trusted Platform Module (TPM) is a hardware chip embedded in platforms that can carry out various cryptographic functions. TPM has a set of

[1] www.google.com/dashboard

special volatile registers called *platform configuration registers* (PCRs). These 160-bit long registers are used to keeping track of the integrity information during a bootstrap process. The TCPA specification defines a set of functions for reporting PCR values [19,20]. When the TPM reports the values of the integrity metrics that it has stored, the TPM signs those values using a TPM identity.

The process of reporting the integrity of a platform is known as *remote attestation*. To achieve the goals of *remote attestation*, TCG has introduced in version 1.1 specifications the concept of privacy certification authority (Privacy CA) [19]. It works briefly as follows. Each TPM is equipped with an RSA key pair called an Endorsement Key (EK). The Privacy CA is assumed to know the Endorsement Keys of all valid TPMs. Now, when TPM needs to authenticate itself to a verifier, it generates a second pair of RSA key called an Attestation Identity Key (AIK), it sends the AIK public key to the Privacy CA, and authenticates this public key w.r.t the EK. The Privacy CA will check whether it finds the EK in its list and, if so, issues a certificate to the TPM's AIK key. The TPM can then forward this certificate to the verifier and authenticate itself w.r.t. this AIK. A user may lose his anonymity in this scheme if the privacy CA and the verifier collude.

As discussed by Brickell, Camenisch and Chen [10], version 1.2 of the TCG specifications incorporate the Direct Anonymous Attestation (DAA) protocol. This protocol is designed to address anonymity issues of remote attestation. In such a scenario, the server only learns that the platform is trusted, but not the identity of the platform. The DAA protocol offers user-controlled linkability.

1.4 Contribution

We introduce in this paper a Cloud Service Architecture based on the Direct Anonymous Attestation Scheme. The key features of the proposed service architecture are:

- A user is able to link Cloud Service applications in such a way, that his/her personal data are shared only among the designated application group.
- The service provider has better assessment and control of a user's accounts, which leads to efficient security enforcement measures.

The plan of the paper is as follows. Section 2.1 - 2.2 presents a high-level description of the Direct Anonymous Attestation (DAA) scheme. Some of the technical features of DAA are explained in section 2.3 - 2.5. The DAA-enabled Cloud Service Architecture is presented in section 3. Section 3.1 - 3.4 explains the components of this architecture and related implementation issues.

2 DAA Protocol Overview

The Direct Anonymous Attestation (DAA) scheme [10] draws upon techniques from the Camenisch-Lysyanskaya (CL) signature scheme [13], identity escrow

and credential systems. The protocol allows remote attestation of a trusted platform while preserving the privacy of the system user. We outline below the important features of the DAA protocol. A more comprehensive description of the DAA scheme can be found in [20,10].

The DAA scheme is made up of two sub-protocols: *DAA join* and *DAA sign*.

2.1 DAA Join Protocol

The Join protocol enables the Host/TPM to obtain a DAA certificate from the DAA issuer.

Let (n, S, Z, R) be the DAA issuer public key, where n is an RSA modulus, and S, Z, R are integers modulo n. We assume that the platform (TPM) is already authenticated to the DAA issuer via its Endorsement Key, EK.

The TPM first generates a secret value f, and constructs the blind message $U := R^f S^{\nu'} \bmod n$ where ν' is a "blinding" value chosen randomly. The TPM also computes $N_I = \zeta_I^f$, where $\zeta_I = (hash(1||bsn_I))^{(\Gamma-1)/\rho} \pmod{\Gamma}$, and Γ, ρ are components of DAA issuer's public key. The TPM then sends (U, N_I) to the DAA issuer, and convinces the DAA issuer that U and N_I are correctly formed (using zero knowledge proof). If the DAA issuer accepts the proof, it will sign the blind message U, by computing $A = \left(\frac{Z}{US^{\nu''}}\right)^{1/e} \bmod n$, where ν'' is a random integer, and e is a random prime. The DAA issuer then sends the TPM the triple (A, e, ν''), and proves that A was computed correctly. The DAA certificate is the then $(A, e, \nu = \nu' + \nu'')$.

2.2 DAA Sign Protocol

The *sign protocol* allows a platform to prove to a verifier that it possesses a DAA certificate, and at the same time, to sign and authenticate messages. The TPM signs a message m using its DAA secret f, its DAA certificate, and the public parameters of the system. The message m may be an Attestation Identity Key (AIK) generated by TPM, or an arbitrary message. If m is an AIK, the key can be later used to sign PCR data or to certify a non-migratable key. In addition, the TPM computes $N_V := \zeta^f \bmod \Gamma$ where ζ is random or derived from the DAA verifier's basename depending on the anonymity requirement (see section 2.3). The value N_V allows for rogue tagging. The output of the *sign protocol* is known as the DAA Signature, σ.

The verifier verifies the DAA signature σ. The verifier needs to be convinced that the TPM has a DAA certificate (A, e, ν) from a specific DAA issuer. This is accomplished by a zero-knowledge proof of knowledge of a set of values f, A, e, and ν such $A^e R^f S^\nu \equiv Z \pmod{n}$. Further it needs to be shown that a message m is signed by the TPM using its DAA secret f, where f is the same value in the DAA certificate.

2.3 Variable Anonymity

The DAA protocol provides user-controlled anonymity and linkability. Precisely, a platform/TPM can achieve the following two statuses: 1. verifier linkable trans-

action 2. verifier non-linkable transaction. The statuses are controlled by the parameter: ζ. If non-linkable transactions are desired, a different and random value of ζ should be used for every interaction with a verifier. If linkable transactions are desired, ζ should be selected from a static basename based on the verifier, e.g $\zeta = (hash(1||bsn_V))^{(\Gamma-1)/\rho} \pmod{\Gamma}$.

2.4 Rogue Tagging

The DAA protocol has a built-in rogue tagging capability. A rogue TPM is defined when its secret value f has been extracted. Once a rogue TPM is discovered, the secret f values are distributed to all potential issuers/verifiers who add the value to their rogue-list. Upon receiving N_V, the verifier can check if N_V is equal to $\zeta^{\tilde{f}}$ for all $\tilde{f}$ stemming from rogue TPMs, and hence tag the TPM if necessary.

2.5 A Privacy Flaw Involving Corrupt Administrators

It is shown in [26] that an issuer and verifier can collude to break the anonymity of the user when linkable transactions are used. This privacy violation relies on the assumption that an issuer and a verifier share the same basename (i.e. $bsn_I = bsn_V$). The authors in the same paper suggest the following security fix. In the Join Protocol, compute $\zeta_I = (hash(0||bsn_V))^{(\Gamma-1)/\rho} \pmod{\Gamma}$. And in the Sign Protocol, compute $\zeta = (hash(1||bsn_V))^{(\Gamma-1)/\rho} \pmod{\Gamma}$. Now this leads to $N_I \neq N_V$ regardless $bsn_I = bsn_V$. So the issuer and the verifier can not identify the user by matching the values N_I and N_V.

3 DAA-Based Cloud Service Architecture

The proposed Cloud Service Architecture is displayed in Figure 2. For simplicity, our discussion is restricted to a single Cloud Service provider who acts both as an issuer and verifier. The user-privacy is preserved even when the Cloud Service

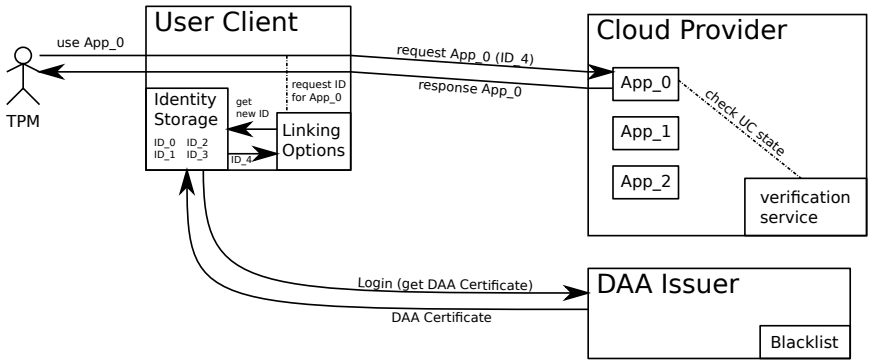

Fig. 2. Privacy-Enhanced Cloud Service Architecture

decides to employ another service for issuing DAA certificates (see section 2.5). A DAA certificate is issued when a user registers/login an account. The credential of the user is checked at this stage following the DAA Join protocol. After the login, a user is able to set linkability options (section 3.2) among the services offered by the Cloud. When a user requests the usage of a particular service, the permission is granted (or denied) by the Cloud Service acting as a verifier after further security analyses (see section 3.4). We discuss below in details the various components of this architecture and some involved implementation issues.

3.1 Service Login

A user is required to login in order to use the services offered by the Cloud. A successful login provides the user a one-time DAA certificate, so that the user can proceed to request further services. Figure 3 shows the behind-scene of a login session for a TPM based platform. The user login session is implemented based on the DAA Join protocol, and since ζ_I is constant (derived from the Cloud Service basename), a rogue-list can be computed and kept afresh at desired time interval. As soon as a TPM is compromised and its secret key f distributed, the service provider can tag the rogue TPM by augmenting the rogue-list. The tagged TPM will not be able to access the account at next login session.

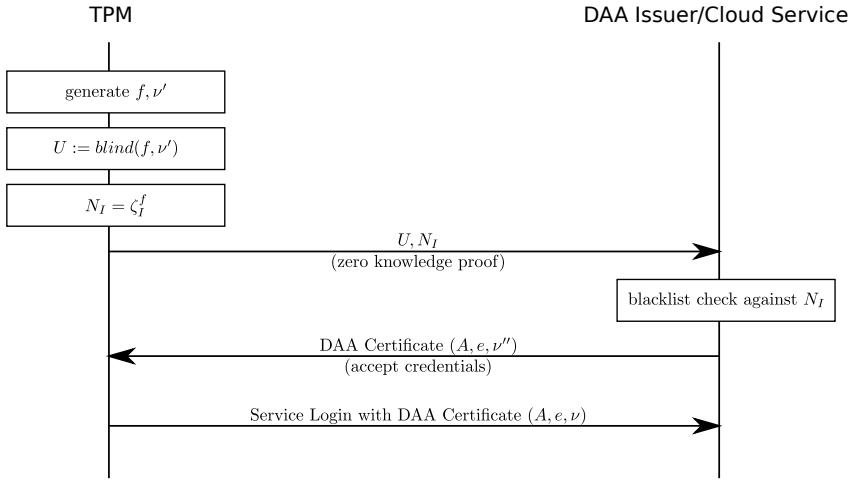

Fig. 3. Cloud Service Login for a TPM

3.2 User-Controlled Linkablility

Upon a successful login, a user is allowed to set linking options among the application services offered by the Cloud. By *Linkable* Applications, we mean that given two or more of service requests/usages originating from the same user, the cloud service is not able to link them and conclude they originate from the

same user. Of course, linkbility has meaning only when anonymity is achieved. Anonymous attestation is achieved at the Service Login stage as explained in the previous section. There are three service link-statuses. Figure 4 shows a typical application linking scenario.

1. **Non-linkable Application.** Service is not able to link a user's transactions
2. **Single Application Linkability.** Service is able to link a user's transactions and data in a single application
3. **Multiple Applications Linkability.** Service is able to link a user's transactions and data across the application group

The linkability mechanism hinges upon how a basename is selected (see section 2.3). For non-linkable applications, a random basename (random ζ) is used during the service transactions. Typical non-linkable applications could be areas where a user performs web searches/browsing, and the user wishes to keep anonymous his search content and also his search history non-linkable. Linkable applications employ a static basename. In particular, each application group (services a user wishes to link) employs a basename which should reflect the service content of the application group (see section 3.3).

The decision as to what service applications to link is of course a personal one. For example, some users may prefer to link services that contain their financial data (credit card number, investment portfolio, etc.). Some limited service users may choose not to link any of the registered services. A Cloud Service should be able to track a user's activities in particular accounts. The history information contained in an account is necessary for purposes such as billing, or further

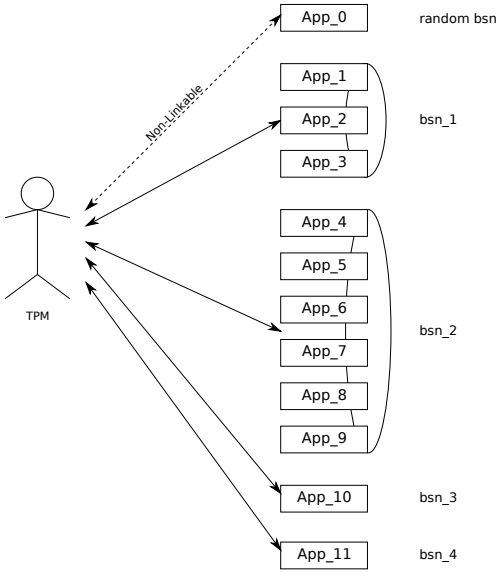

Fig. 4. User-Controlled Linkable Cloud Service Applications

service upgrading. On the other hand, by restricting information sharing among the service accounts, a user can be assured his/her information when divulged will only contain those parts of the profile.

3.3 Selecting a Basename

The linkable applications share a common basename. The basenames must be available as soon as a user sets the linking configuration, and before requesting service usages. The current DAA scheme does not provide protocol procedures specifying how basenames should be generated. One possible solution is to pre-generate a list of basenames containing all possible combinations of linkable applications. This list is stored on the server and becomes available once a user sets a particular linking configuration. However, this solution may become impractical as the number of services increases (leads to an exponential growth of options). A more efficient solution might be to create a basename generating program. Whenever a user sets the linking configuration, suitable basenames can be generated and saved on the servers. Smyth et als. [26] discussed some alternative approaches in constructing and managing the basename lists.

3.4 Account Suspension/Closing

DAA rogue tagging capability allows Cloud Service providers to suspend or close a user's account when they see suspicious behavior on the part of a user. The specifics of the rogue behavior is of course application dependent. For example, in an application involving software download/upgrade, the Cloud Service may require that remote platform to prove its trustworthiness by providing the platform's PCR values. The failure of compliance or unsatisfactory reporting results on the user part may lead to a rogue status. Other suspicious behavior could be: above-normal usage of a particular account, repeated account creation and deletion, and any other discretionary rules decided by the verifier.

Also since a common basename (constant ζ) is used among linked-applications, the Cloud Service (as a verifier) is able to update the rogue list regularly. The DAA Sign protocol can be efficiently carried out using batch proof and verification techniques [14,8]. In fact, Chen's asymmetric pairing based Sign Protocol [14] is extremely computationally efficient. For each signing process, the TPM is only required to compute one exponentiation if linkability is not required, and two exponentiations when linkability is required. The efficiency of this scheme comes from an ingenious use of a batch proof and verification scheme in proving the discrete logarithmic equality between two group elements y_1 and y_2 to two bases g_1 and g_2 respectively (i.e. $\log_{g_1} y_1 = \log_{g_2} y_2$) .

4 Related Work

The DAA scheme is introduced in the seminal paper [10]. The DAA protocol is designed to address anonymity issues of remote attestation. Privacy flaws

were found after the introduction of the original protocol [25,26] and the corresponding security fixes are suggested in [24,26]. There have been work done to enhance capabilities of the original DAA scheme. Camenisch [12] suggested a hybrid anonymous attestation scheme which combines the DAA and the privacy CA approaches. Brickell and Li [11] introduced a new DAA scheme called *Enhanced Privacy ID*. The new DAA scheme while providing non-linkability, is capable of revoking a TPM even if the TPM private key is unknown.

Data security and privacy is one of the biggest challenges in Cloud Computing. Cloud data must be protected not only against external attackers, but also corrupt insiders (administrators). The *information-centric* approach [5,17] aims to make cloud data self-intelligent. In this approach, cloud data are encrypted and packaged with a usage policy. The data when accessed will consult its policy, create a virtualization environment, and attempt to assess the trustworthiness of the data environment (using Trusted Computing).

Applied Cryptography offers tools to address privacy and security questions related to cloud data. Recent advances in cryptography allow remote operations, manipulations and computations on encrypted cloud data. The predicate encryption scheme [27,9] allows cloud based searches on encrypted documents. Homomorphic encryption [18,28] and Private Information Retrieval (PIR) [15] can perform computations on encrypted data without decrypting.

To make cloud services more secure and reliable, Google has launched a prototype hardware **Cr-48**, which is tailor-designed to run the Google Chrome Operating System [1]. The prototype hardware is shipped with Trusted Platform Modules. About 60,000 Cr-48s were manufactured and distributed to testers and reviewers in early December 2010. Reviews published in mid-December 2010 indicated that while the project holds promise, it is still not market-ready [29]. In the EU framework, **PrimeLife** [6] is an ongoing research project funded by the European Commission. The main objective of the project is to bring sustainable privacy and identity management to future networks and cloud services.

5 Conclusion

Cloud Service is clearly becoming one of today's most popular Internet-based services, due to its cost-efficiency and flexibility. The future development of Cloud Services relies on mature technology deployment in areas, such as hardware/software security, data provision infrastructure, and reliable Third-party data control [16].

To better protect user data, we have in this paper introduced a cloud service architecture based on the Direct Anonymous Attestation scheme as outlined in the Trust Computing Group specification [20]. The theoretical DAA-based architecture provides cloud users the abilities of controlling the extent of data sharing among his service accounts, proving the integrity of his platform (PCR values) to remote servers, and the most important of all, preserving anonymity of the platform identity.

Acknowledgements. Many thanks go to the anonymous reviewers who provided detailed and insightful commentary on this paper.

References

1. Chrome Notebook, http://www.google.com/chromeos/pilot-program-cr48.html
2. Google Buzz, http://www.google.com/buzz
3. Google Latitude, http://www.google.com/latitude
4. Google Transparency Report, http://www.google.com/transparencyreport/governmentrequests/
5. An information-centric approach to information security, http://virtulization.sys-con.com/node/171199
6. The Primelife Project, http://www.primelife.eu/
7. Warning: Google buzz has a huge privacy flaw (February 2010), http://www.businessinsider.com/warning-google-buzz-has-a-huge-privacy-flaw-2010-2
8. Bellare, M., Garay, J.A., Rabin, T.: Fast Batch Verification for Modular Exponentiation and Digital Signatures. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 236–250. Springer, Heidelberg (1998)
9. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public Key Encryption with Keyword Search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004)
10. Brickell, E.F., Camenisch, J., Chen, L.: Direct Anonymous Attestation. In: ACM Conference on Computer and Communications Security, pp. 132–145 (2004)
11. Brickell, E., Li, J.: Enhanced Privacy ID: a direct anonymous attestation scheme with enhanced revocation capabilities. In: WPES, pp. 21–30 (2007)
12. Camenisch, J.: Better Privacy for Trusted Computing Platforms (Extended Abstract). In: Samarati, P., Ryan, P.Y.A., Gollmann, D., Molva, R. (eds.) ESORICS 2004. LNCS, vol. 3193, pp. 73–88. Springer, Heidelberg (2004)
13. Camenisch, J.L., Lysyanskaya, A.: A Signature Scheme with Efficient Protocols. In: Cimato, S., Galdi, C., Persiano, G. (eds.) SCN 2002. LNCS, vol. 2576, pp. 268–289. Springer, Heidelberg (2003)
14. Chen, L.: A DAA Scheme Using Batch Proof and Verification. In: Acquisti, A., Smith, S.W., Sadeghi, A.-R. (eds.) TRUST 2010. LNCS, vol. 6101, pp. 166–180. Springer, Heidelberg (2010)
15. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. J. ACM 45(6), 965–981 (1998)
16. Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R., Molina, J.: Controlling data in the cloud: outsourcing computation without outsourcing control. In: Proceedings of the 2009 ACM Workshop on Cloud Computing Security, CCSW 2009, pp. 85–90. ACM, New York (2009)
17. EMC. Information-centric security, http://www.idc.pt/resources/PPTs/2007/IT\&Internet_Security/12.EMC.pdf
18. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, pp. 169–178. ACM (2009)
19. Trusted Computing Group. Trusted computing platform alliance (TCPA) main specification, version 1.1b (2001), www.trustedcomputing.org

20. Trusted Computing Group. Trusted computing platform alliance (TCPA) main specification, version 1.2 (2003), www.trustedcomputing.org
21. Privacy International. An interview with google on government access to personal information, https://www.privacyinternational.org/article/interview-google-government-access-personal-information
22. Privacy International. Privacy international identifies major security flaw in google's global phone tracking system, https://www.privacyinternational.org/article/privacy-international-identifies-major-security-flaw-google's-global-phone-tracking-system
23. Lambert, C.: Google latitude, now with location history and alerts (November 2009), http://googlemobile.blogspot.com/2009/11/google-latitude-now-with-location.html
24. Leung, A., Chen, L., Mitchell, C.J.: On a Possible Privacy Flaw in Direct Anonymous Attestation (DAA). In: Lipp, P., Sadeghi, A.-R., Koch, K.-M. (eds.) Trust 2008. LNCS, vol. 4968, pp. 179–190. Springer, Heidelberg (2008)
25. Rudolph, C.: Covert Identity Information in Direct Anonymous Attestation (DAA). In: SEC, pp. 443–448 (2007)
26. Smyth, B., Ryan, M., Chen, L.: Direct Anonymous Attestation (DAA): Ensuring Privacy with Corrupt Administrators. In: Stajano, F., Meadows, C., Capkun, S., Moore, T. (eds.) ESAS 2007. LNCS, vol. 4572, pp. 218–231. Springer, Heidelberg (2007)
27. Song, D.X., Wagner, D., Perrig, A.: Practical techniques for searches on encrypted data. In: IEEE Symposium on Security and Privacy, pp. 44–55 (2000)
28. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully Homomorphic Encryption Over the Integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010)
29. Wolfgang, G.: Chrome OS is ahead of its time (December 2010), http://www.conceivablytech.com/4624/products/chrome-os-is-ahead-of-its-time