Enforcing Privacy in E-Commerce by Balancing Anonymity and Trust

Giampaolo Bella, Rosario Giustolisi, Salvatore Riccobene

Dipartimento di Matematica e Informatica
Università di Catania, ITALY

Abstract

Privacy is a major concern in e-commerce. There exist two main paradigms to protect the customer's privacy: one relies on the customer's trust that the network will conform to his privacy policy, the other one insists on the customer's anonymity. A new paradigm is advanced here as a natural balance between these two. It sees the customer act using his real identity but only circulate cover data that conceal the resources he requires. Privacy enforcement is thus shifted from the customer's identity to his purchase preferences. The new paradigm is suitable for scenarios such as eBay purchases where trust that a network sticks to a privacy policy is problematic, while anonymity is either forbidden or impossible.

The computation of cover data is done by a node other than the customer in order to minimize impact on the customer. That node will therefore see the customer's private data that are used to compute the cover. This demands some technology to prevent the node from exposing private data. An existing protocol developed for self-enforcing privacy in the area of e-polls is thoroughly analysed and found somewhat weak in terms of fairness among its participants. A stronger version is designed and adopted, together with an innovative differential-privacy preserving function, in the new privacy paradigm. The strengthened e-poll protocol and the new differential-privacy preserving function, which strictly speaking only are side contributions of this paper, each appear as important as the new e-commerce privacy paradigm.

Keywords: Self-enforcing privacy; differential privacy; customer privacy; security protocol, e-polling; pollster.

Contents

1. Introduction

Privacy is often erroneously abused as confidentiality, although it rather indicates a right to confidentiality, that is "the right of an individual to decide when and how sensitive personal data should be revealed" [1]. For example, a customer's personal data are private in the EU so that he can decide whether or not to disclose them. The consequences of abusing private data such as people's purchase preferences are well known to the marketing industry. However, ensuring that a customer's data are kept private while they traverse a network of computers is far from trivial. Most security protocols appeared in the last three decades, from Kerberos [2] to SSL/TLS [3], only aim at transmitting data confidentially but assume that the initiator is willing to share them with the responder. That assumption is not met when privacy is required.

This is often the case with e-commerce, for example. In a typical transaction, a customer contacts a node to obtain some desired resources, and enters his personal data. The node may be unable to honour the request on its own, and hence may need to variously collaborate with others. This requires the sharing of (some of) the customer's personal data, such as what he wants to buy. Finalising the purchase then requires the selling node to interact with a bank to manage the payment, and with a shipment society to deliver the purchased items. The result is that the customer's private data have flown through several nodes, each handling them according to its own privacy policy, while the customer usually remains uninformed of the various branches or paths the flow may develop in.

There exist two main privacy paradigms in e-commerce. One rests on the customer's trust that the network conforms to his privacy policy, so that he accepts to transmit his identity and required resources, but is able to suspend or resume trusted nodes' treatment of his personal data. The other one is based on anonymity, so that data are linked with a pseudonym and not with the real customer's identity. We advance an e-commerce privacy paradigm as the natural tradeoff between these two. There are many real-world scenarios, such as eBay shopping, where a customer may not sufficiently trust the network to handle his data privately, and at the same time anonymity is either forbidden or impossible due to unavailability of a privacy certification authority (which stores the pseudonyms). Our paradigm applies here because it removes both the need for the customer's trust in the network and the need for his anonymity.

The gist of the new paradigm is that the customer uses his real identity but only circulates data that cover the actual resources he is looking for — this is the data concealment phase. Such data will be orchestrated through the network to raise potential matchings, and each node will use certified e-mail to send the customer a matching offer in a standardised format for mechanical processing — this is the orchestration phase. The customer will only disclose the very resource names, via a fair-exchange scheme, to the node he has chosen, so that the customer's trust reduces from the entire network to only this end node — this is the completion phase. The three phases are as light-weight as possible upon the customer so that they can be easily implemented as a *privacy-preserving e-commerce service* to run on the customer's machine in order to

automate the interaction with the customer. He would only have to enter his required resources and wait for the best-matching offers.

Data concealment turns out the most complicated phase. Our main requirement was to relieve the customer, who might be a casual customer, from the burden of calculating cover data. Also, these would be more suitably calculated by a merchant node according to the most appropriate business rules. However, meeting this requirement would raise the risk of privacy breach because the resources that the customer requires should be handed over to a participating node in an intelligible form, for the node to calculate cover data. We were pleased to find in the area of e-polls some technology that would contribute meeting our two-faceted requirement. It is a protocol that we call SEP [4], aimed at self-enforcing privacy, which we deeply scrutinised in a strong threat model. Our analysis revealed a conceptual weakness in the protocol in terms of insufficient fairness, and inspired our upgraded, stronger protocol, called SEP+. E-commerce clearly is a broader application than e-polls, hence our paradigm continues with the orchestration and completion phases, which however turn out rather elaborate but not controversial.

The contribution of our research therefore is at least threefold. Its main one is a privacy paradigm that balances anonymity and trust in e-commerce. What was born as a side contribution, but in the end turned out to be at least as relevant as the main one, is the thorough analysis of the SEP protocol and the design of its strengthened version SEP+. This paper, which is about the design and informal analysis of our technology, conjugates the findings of two conference papers [5, 6]. However, it entirely rewrites them upon the basis of significant extensions to both the design and the analysis aspects. A major extension, and third contribution of this paper, is the definition of what seems to be the first differential-privacy preserving function with non-numeric values, which is used to simulate the main phase of our paradigm.

Our treatment begins with protocols for self-enforcing privacy, presenting SEP, its analysis and SEP+ (§2). It moves on to privacy enforcement in e-commerce by outlining the two main existing paradigms (§3). Then, it describes our e-commerce privacy paradigm (§4). After that, it introduces our differential-privacy preserving function with non-numeric values, and uses it to simulate the main phase of our paradigm (§5). Finally, it makes some discussion (§6), and terminates (§7).

2. Protocols for Self-Enforcing Privacy

We call SEP a recent protocol to release data with self-enforcing privacy; it is the final and main one in a triple where the first allows no release of data and the second only allows a randomised response [4]. A weakness was recently found in the first protocol of the triple [7], and thus is not directly related to our work. This Section presents SEP (§2.1), continues with our analysis of the protocol (§2.2) and terminates with SEP+, our strengthened protocol (§2.3).

2.1. The SEP Protocol

Golle et al. [4] develop an e-poll protocol to trace data after transmission. As depicted in Figure 1, each individual is required to add to his preferences P_1, P_2, ..., P_{p_1} some information that must link the preferences with the pollster. This additional information B_1, B_2, ..., B_{b_1} serves as baits. Therefore, each individual in fact transmits a bundle containing his preferences and the baits.

It is important that the baits do not compromise the results of the poll, and the pollster be unable to distinguish whether the received bits are actual preferences or baits. If the pollster is dishonest and publishes or sells individuals' private data, then the individuals must be able to indict it publicly. The indictment is possible exactly because the published data contain the individuals' baits. Like the authors of SEP, we are not concerned here with how the individuals can practically realise that their data have been published, a process termed *leak return* [4, Fig.1].

It is also necessary to ensure that the pollster cannot be indicted illicitly. Since the individuals have the baits, they could insert them in a fake data collection, then publish the collection, and finally indict an honest pollster. A fair scheme must make such an indictment impossible.

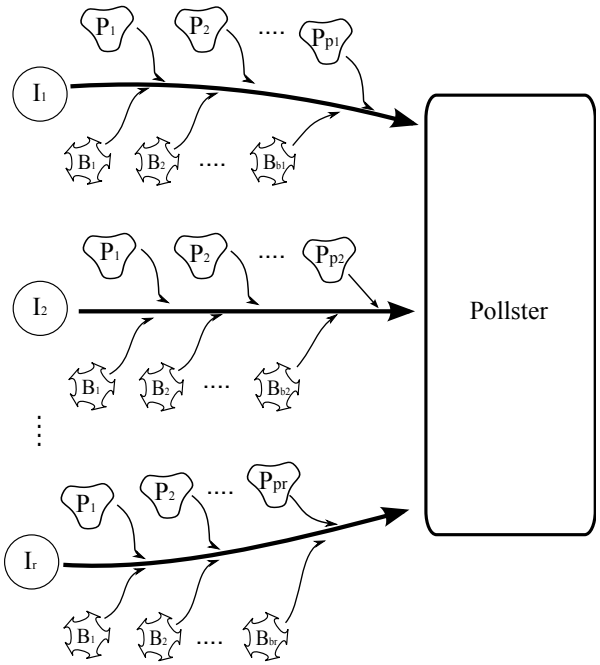

Figure 1: The original SEP protocol

The SEP protocol attempts to implement a fair scheme of preference submission. It adopts the RSA encryption scheme to endow the pollster with a public encryption function E and a corresponding private decryption function D. It

is understood that anyone can apply E as it requires the pollster's public key, whereas only the pollster can apply D as it requires its own private key. The main idea is that each individual computes the baits using a hash function that the pollster makes public. That function is required to have as image the set of ciphertexts that can be produced using the underlying cryptosystem. As a result, the pollster will be unable to discern whether a ciphertext was produced using the hash function, in which case it is a bait, or using the actual encryption algorithm, in which case it is a real preference.

We can now move on to describe the protocol in detail. It is composed of four main phases plus the indictment phase.

- ***Setup.*** The pollster publishes the parameters for the encryption algorithm E and two hash functions. One, named h, is a standard hash such as SHA-256; the other one, named g, is a special hash function whose image is the same as the encryption function's, that is $\mathcal{I}m(g) = \mathcal{I}m(E)$.
- ***Sending a bit to the pollster.*** The individual's preference is sent bit by bit. To send a bit $b \in \{0, 1\}$, the individual whose identity is I_i chooses a random value r such that the least significant bit of $h(I_i \parallel r)$ is b. The individual sends $I_i, E(r)$ to the pollster.
- ***Sending a bait to the pollster.*** To send a bait to the pollster, the individual chooses a random value s, computes $g(s)$ and sends $I_i, g(s)$ to the pollster.
- ***Decryption.*** Given an identifier I_i and a ciphertext c, the pollster decrypts c to recover the plaintext p, then computes the least significant bit b of $h(I_i \parallel p)$ and stores it. Such b is a single bit of I_i's preference.

Because E and g have the same image, the pollster cannot discern which function was used to build the ciphertext c it receives. It can only decrypt it as described above, obtaining a plaintext p. Only if c was computed using E do we have that $p = r$. On the other hand, the individual cannot later decrypt $g(s)$, that is calculate $D(g(s))$, because he ignores the appropriate key, and hence cannot predetermine the bait that the pollster will decrypt.

However, although the pollster is not entitled to publish the preferences as such, it must be allowed to publish, without a significant risk of indictment, some data about them. Such data must be computed using a function that conforms to the definition of *ϵ-differential privacy* [8] (Definition 1).

Definition 1 (ϵ-differential privacy). *A randomized function f over data sets gives ϵ-differential privacy if for any two data sets X_1 and X_2, which differ in at most one point, and $S \subseteq Range(f)$, the following holds:*

$$\Pr[f(X_1) \in S] \leq \exp(\epsilon) \times \Pr[f(X_2) \in S]$$

The intuition behind this Definition is simple. We may think of X_1 and X_2 as two databases that differ in only one record. A function f satisfies the

definition of ϵ-differential privacy if, once fixed a small value of the ϵ parameter, there are similar probabilities that the respective applications of f to the two databases yield the same "feature" S. Conversely, for relatively big values of ϵ, those probabilities may differ significantly. We remark that the ϵ parameter is typically omitted in informal prose although, as we shall see, it plays an important role.

A chief property for the pollster is that the computation of the function eliminates the baits so that, by publishing its results, the pollster will expose insufficient information for the individuals to indict it. As an extreme, if the pollster used the constant function $f(x) = 1$, it would be on the safe side because the constant reveals no baits, but such a function would be much too inaccurate. Conversely, if the function also satisfies ϵ-differential privacy, then its output can be tuned towards concealment of the input, that is privacy, by means of small values of ϵ, or towards accuracy by means of relatively big values of ϵ. We provide a demonstration below (§5). Many useful functions, such as individual component analysis and k-means clustering, can be constructed to be differential-privacy preserving [9]. In general, a function can be made differential-privacy preserving by adding laplacian noise or by the exponential mechanism [10].

In our application, we may think of the example set X_1 as the set of clean preferences and of the example set X_2 as the set of preferences enriched with the baits. A dishonest pollster may publish the raw collected data (containing the baits), thus breaching the privacy of the individual who submitted them. If this is the case, that individual can start the indictment phase thanks to the clues that the baits provide. To succeed, the individual must show a number of valid exhibits of the form:

$$I_i,\ s_i,\ b_i$$

where I_i is the individual's identity, s_i is the bait and b_i is the indicted bit. Recall that D denotes the decryption function, which only the pollster can apply, corresponding to E, which anyone can apply. The judge will deem an exhibit valid *if and only if* the least significant bit of $h(I_i \parallel D(g(s_i)))$, which only the pollster can compute by applying D, is equal to b_i. The security analysis of this phase will be crucial (§2.2).

Two parameters are important to regulate the indictment phase and therefore should be pre-agreed out of band between the pollster and the participating community. One, indicated as n_0, is the validity threshold for the accusation. The individuals shall then advance a number of valid exhibits higher than n_0 to successfully accuse the pollster. It is interesting that different individuals can contribute to reaching that number, precisely those whose data the pollster putatively published. Since each exhibit is based on the value of a single bit, an individual might just be successful at guessing a valid exhibit. But n_0 reduces the probability that the individuals guess a sufficient number of valid exhibits to $\frac{1}{2^{n_0}}$.

Another important parameter is indicated as w_n and represents a sort of verdict's tolerance. The pollster can successfully contest the indictment by demon-

strating that at least $(\frac{1}{2} - w_n)n$ of the alleged exhibits are invalid. Therefore, the pollster will need to invalidate as fewer than half the exhibits as defined by w_n. It proves that an exhibit is invalid by outputting $r_i = D(g(s_i))$, with a proof of correct decryption, and demonstrating that the least significant bit of $h(I_i \parallel r_i)$ is not b_i.

The ϵ parameter in Definition 1 is linked to n_0 and w_n. More precisely, "safe values of ϵ in turn depend on the values of n_0 and w_n that govern the indictment rules. These values must be chosen to permit a sufficient level of safe disclosure" [4, §5].

2.2. Evaluating SEP

Let us consider the following real-world scenario for the sake of evaluation. A pollster P claims an investigation about the people who are interested in a life insurance contract. The individuals submit their preferences bundled with baits. Then, P collects the preferences, applies its chosen function that conforms to Definition 1. Finally, it publishes the output.

Although everyone has behaved honestly so far, it may be the case that, after the publication of the results, some malicious individuals decide to accuse P for an unfounded privacy breach, aiming in fact at a refund for a hypothetical violation. Also the opposite violation is possible, as a malicious pollster may purposely breach the individuals' privacy by selling the clean collected preferences to an insurance company.

SEP must be evaluated in the real world, where it must be assumed that anyone may seek personal benefit by acting maliciously. Because the protocol is not deployed, this issue can only be addressed by abstract analysis. It is well known that only an abstract analysis that relies on some formal, mathematically grounded, method can provide a rigorous evaluation. Despite our experience in formal protocol analysis [11], we found out that even an informal analysis of SEP denounces that the protocol has room for improvement, as detailed below.

The most appropriate threat model to evaluate SEP appears to be *BUG* [12], which is named as a permuted acronym of "The Good, the Bad and the Ugly". It partitions the protocol participants into three sets according to their behaviour:

- *good* nodes always conform to the protocol;
- *bad* nodes always attempt to break the protocol for the sake of personal profit;
- *ugly* nodes may, in turn, resort to either good or bad behaviour; they may, either deliberately or not, favour bad nodes.

It seems fair to observe that BUG has opened the ground to new findings about security protocols, which historically had exclusively been studied against the single super-potent attacker theorized by Dolev and Yao [13]. Chiefly, BUG allows scenarios in which nodes do not collude but work for personal profit instead. Hence, it is most appropriate to studying SEP realistically. Each

static picture of the network, depicting the nodes with the messages they have sent or received up to that stage, can be characterized in terms of behaviour. For example, Lowe's famous attack sees the man in the middle acting as bad, the end point as good and the initiator as ugly [12]. However, nodes may change behaviour, so that other pictures may show a different partition. More details about BUG can be found in its dedicated publication [12] or in its recent simplification and mechanization in a model checking tool [14].

Before the actual protocol evaluation, it is important to clarify the meaning of an exhibit. An exhibit may be seen as a claim of an individual's. If the individual is good and the pollster is bad, then the individual found the bit b_i illicitly published somewhere by the pollster; viceversa, he just guessed b_i. The individual's claim is that b_i belongs to himself because it is computationally linked to a bait he can exhibit (that is s_i) only after application of the decryption algorithm publicly associated to the pollster. The truth value of this claim can be easily verified by decrypting the bait as detailed in the previous Section. Intuitively, if the individual is good (and therefore truthful) and the pollster is bad, then with high probability the claim will be verified as true, otherwise it will be verified as false. Thus, if the individuals who submit exhibits are good and the pollster is bad, then the minimum exhibit threshold n_0 will be reached.

We can now analyse SEP, informally though systematically, from the standpoints of the individuals or of the pollster. The analysis shows how the protocol counters scenarios with various behaviours.

The individuals. If they are good, then they submit their preferences correctly bundled with the baits, and attempt no dispute. If they are bad, they may attempt to build by themselves a fake collection of preferences with baits and publish it. Then, they may build a number of exhibits that will generally turn out invalid, and the pollster will get by thanks to the minimum exhibit threshold n_0 and the verdict's tolerance parameter w_n explained above (§2.1). If the individuals are ugly, then they may decide, according to their personal cost/benefit analysis, to take a good behaviour at times and a bad behaviour at other times. Still, the pollster cannot be indicted if it followed the protocol because the exhibits against him are insufficient.

The pollster. If the pollster is good, it does not commit any privacy breach because it publishes the output of a differential-privacy preserving function. Therefore, the individuals cannot indict it. If the pollster is bad, it may collect the individuals' preferences and then publish or sell them to a third party for example. Then, the individuals may start the indictment phase but the pollster may decide not to show up at court, or pretend a technical problem such as a denial-of-service attack if the indictment were to take place remotely via the Internet, and only participate partially. Without the pollster's complete participation, none or an insufficient number of proven correct decryptions of the baits can be produced, so that the individuals will fail to reach a sufficient number of valid exhibits. Hence,

the pollster will go unpunished! Finally, if the pollster is ugly, it may over time give rise to either one of the scenarios described here.

Our evaluation confirms that SEP is robust against the individuals' malicious behaviour, which is an important feature. However, it also reveals that when the node that acts maliciously is the pollster, the individuals are unable to indict it because they have none or an insufficient number of valid exhibits.

It is somewhat surprising that SEP requires the pollster to collaborate in its own indictment (by decrypting, with a proof of correct decryption, the baits) because it seems highly unlikely that a dishonest pollster would help its own accusation in practice. Most importantly, criminal trials in the real world can reach an end even *in absentia* of one of the parties, that is when one of the parties fails to show up at court, whereas SEP's indictment phase cannot without the pollster's participation. Of course, the law might introduce dedicated regulations to bind the pollster to participate, but in such case the technology would be sure to have failed its goal. For example, the Zhou-Gollman protocol ensures non-repudiation of origin and of receipt against false claims without the indictee's contribution in any case [15].

We conclude that SEP as it stands violates the fairness requirement between its peers by giving some advantage to the pollster. The protocol should be strengthened so as to guarantee the individuals' successful accusations even when the pollster refuses to collaborate.

2.3. A Strengthened SEP: SEP+

The previous Section showed that SEP gives some advantage to the pollster over the individuals: because the pollster itself is essential to its own indictment, it can practically avoid being indicted. This is the main motivation to improve SEP. Our aim is to give the individuals evidence that their data were sent to a specific pollster, and to make that evidence sufficient to indict a dishonest pollster without the pollster's contribution. This would provide the required fairness.

We pursue and reach our aim by adopting digital signatures and a complete Public Key Infrastructure (PKI). The pollster is required to be registered with the PKI so that it is equipped with a signature key pair and relative certificates, private signature creation algorithm S and public signature verification algorithm V[16]. By using V, the individuals will be able to verify validity and integrity of data signed by S. More precisely, the individuals will verify the validity of the pollster's certificate, that is the validity of its signature key, through the PKI, and ultimately associate the signed data to a specific pollster. Because a Global PKI is not available at present, our requirement may appear limitative. However, it is not stronger than SEP's requirements of a public encryption algorithm E based on RSA (§2.1) and of a function g, which are both to be associated with a specific pollster. Also these associations required some PKI to let the individuals correctly pinpoint the required pollster. Moreover, secure, global e-polls do not seem an issue at present.

Having described our main and only requirement to strengthen SEP, we can move on to explaining our design upgrades. Our first attempt, which failed as we shall see, also caused the extra requirement that each individual I be registered with the PKI and endowed with a private encryption algorithm E_I and a public decryption algorithm D_I. As for the protocol design, the two sending phases should be augmented with an extra message: when an individual sends real data or baits to the pollster, he should wait for an ack message from it.

The form of the ack message is crucial. It shall deliver to the individual the decryption that he may subsequently need for the indictment, that is $E_I(D(c))$, and shall be signed by the pollster for the sake of integrity and authenticity. A possible ack message would then be of the form:

$$S(I, c, t, E_I(D(c))), Cert_P$$

where I is the individual's identity, t is the current timestamp, $Cert_P$ is the pollster's signature-key certificate, and c is the ciphertext that the individual sends per each bit of preference or per each bait. We remind (§2.1) that c is:

- $E(r)$: if the individual sent a bit of his preference;
- $g(s)$: if he sent a bait.

When the individual receives the ack message, he verifies the validity of the pollster's certificate by contacting the PKI and precisely the very Certification Authority (CA) that signed the certificate. Then, with the right public key available, he verifies the digital signature using the V algorithm. The individual can deduce that something went wrong and abort the session in any of the three following cases: the signature verification fails; the pollster fails to send the ack message; the timestamp is expired. Incidentally, the fact that the pollster is registered with the PKI and precisely with a CA provides a reliable identification mechanism that may help in the subsequent indictment phase.

It is worth remarking that the ack messages provide an individual with the decrypted versions of the data that he previously sent. Therefore, if he sent a bit of preference, then he will receive r; otherwise if he sent a bait, then he will receive the decryption of $g(s)$, which he did not know otherwise. The decrypted value can be read only by that individual, because it is encrypted using E_I.

Also our updated protocol must be evaluated in the BUG threat model, especially to assess whether anyone may have some advantage over anyone else. If the pollster P is bad and publishes or sells the collected data to a third party, then the individuals may start the indictment phase. If P does not participate here, the individuals may collect the decrypted baits from the ack messages and reach a sufficient number of valid exhibits. Therefore, they will be able to indict P all the same. Conversely, if P is good and the individuals are bad, they may make a fake collection using the decrypted versions of the baits, publish that collection and then accuse P. In this case, P has no means to defend itself, and consequently will be unfairly indicted.

It can be noted that this updated protocol removes the advantage from the pollster but moves it on the individuals. In consequence, our updates fail to

make the original protocol fair. It becomes clear that it is excessive to give the decrypted baits to the individuals before any actual indictment. Having learned the lesson, we advance a different update to the protocol, resulting in what we address as SEP+, and achieve more fairness.

We remind that SEP+ assumes a PKI and a registered pollster with signature algorithms V and S, but sets no similar requirement for the individuals. SEP+ extends the original sending phases with a simple ack message (simpler than the previous attempt) of the form:

$$S(I, c, t), Cert_P$$

It can be observed that this message does not provide the decrypted c, that is the pollster simply replies by signing the just-received pair along with the current timestamp. As with the failed variant, an individual will continue the protocol if and only if the pollster sends the acks correctly and timely.

The layout of the protocol is difficult to depict completely. Figure 2 shows the bundled preferences and baits that each individual sends, and also features one ack per individual, which the pollster sends in reply to either a single bit of preference or to a bait of the individual's.

Using timestamps has two obvious drawbacks. One is that all clocks must be synchronised. The other one is that a bad pollster may attempt inserting a more recent timestamp, although this might only convince the individual to bear extra network latency. The known alternative is a nonce round trip, which sees the individual issue a fresh nonce to accompany each message of his, and the pollster quote the same nonce in each reply of his. Although the latter is more robust a mechanism, it also is more computationally demanding. Due to the large number of messages that the individual has to issue, we preferred to opt for the former alternative.

SEP+ passes an analysis against the BUG threat model more successfully than SEP did (§2.2). The analysis is identical except for the case in which the pollster P, who runs SEP+ with the individuals, is bad. This time, as P publishes the collected preferences, the individuals can still indict P even if it does not participate actively in its indictment. They have collected the ack messages that are signed by P. This means that the individuals have evidence that their data where received by P. In particular, P's signature may have been pre-agreed to signify that the pollster accepts compliance with the individuals' privacy policy. Thus, the ack messages qualify as valid clues that the individuals can show to the judge if the pollster committed a breach and then did not want to participate in its own indictment.

Conversely, if the individuals are bad and the pollster P is good, then they will make a fake collection by themselves, publish it and finally attempt to accuse P. Their case will be stronger than with SEP because they can also exhibit the pollster's signed ack messages. Still, the judge will require a sufficient number of valid exhibits, which the individuals cannot build by themselves because they do not know the decrypted versions of the baits, as with SEP. This time, P will be interested in participating because it can prove its honesty: it will decrypt

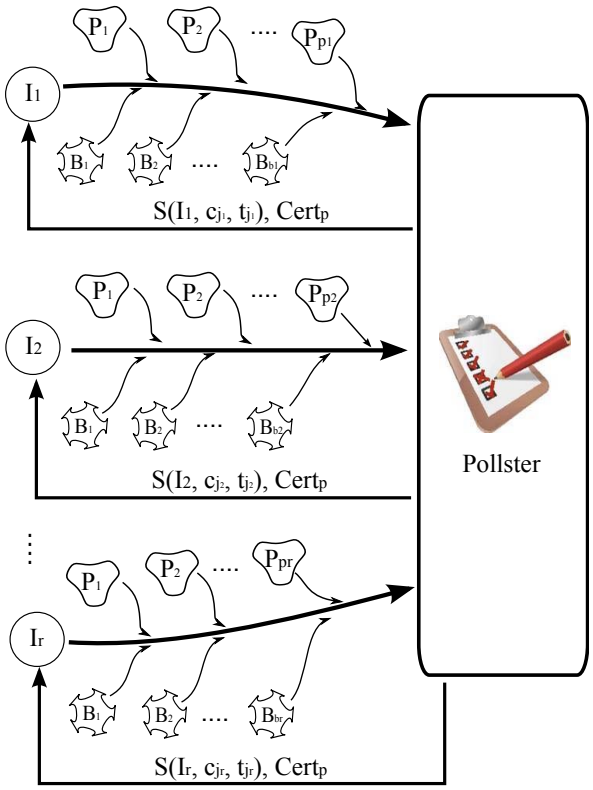

Figure 2: SEP+: our fairer variant of SEP

all baits from the individuals' exhibits showing that the valid ones are fewer than the threshold n_0. In the end, P will not be indicted illicitly.

The indictment phase of SEP+ is similar to that of SEP but is fairer. The individuals have significant evidence against a bad pollster even without the pollster's collaboration, but their evidence is insufficient if they attempt to indict a good pollster. The integrity of the acks ensured by the digital signatures is crucial here. They help tracking the pollster's participation, a useful feature that SEP did not have. In brief, SEP+ narrows down the pollster's malicious behaviour by balancing fairness towards the individuals.

3. Existing Paradigms of Privacy Enforcement in E-Commerce

The aim of this Section is nowhere near an exhaustive presentation of privacy enforcement technologies. By contrast, it is meant as a brief outline of those that seem to be the main paradigmatic approaches to preserve a customer's privacy from a networked audience. One insists on the customer's trusting the audience to preserve his privacy (§3.1), the other one relies on the customer's anonymity from the audience (§3.2). These two paradigms may appear opposite

to each other, hence our idea to define a balance between them, as we shall see later.

3.1. Trust: Suspending and Resuming Data

Waidner and Schunter design a suite of protocols (briefly addressed as WS protocol in the following) to let a customer manage his private data across a network of trusted nodes [17]. Nodes are trusted in the sense that they will conform to the customer's privacy policy, which therefore is the paradigm underlying this protocol. In terms of protocol design, this means that the participating nodes will conform to the protocol without deviating from its prescribed steps. The treatment develops in the context of electronic commerce in the Web 2.0, an "on-line retail scenario" [17].

The WS protocol sees a customer transmit his identity and what he wants to buy to a relevant node in the trusted network, e.g. a bookseller. The bookseller may collaborate with the other nodes in the trusted network in order to fulfill the customer's request. The authors suggest that each node have a privacy panel that allows the customer to manage his data at the collaborating nodes' sites. The customer can view the node privacy policy from its panel and so decide whether it conforms to his own. If the check is affirmative, the customer may decide to release his data to that node, and its privacy panel will then state whom the node disclosed the customer's data to. Another functionality of a node's privacy panel is to let the customer delete or block the node's use of his data.

The customer bundled his data with various ACLs (Access Control Lists) to specify who can do what on them, and with a DF (Data Flow) matrix to indicate his intended flows for the data, that is from which node to which node the data may travel. The ACLs and the DF coming with the data implement the customer's privacy policy on his data. Both these structures are digitally signed by the originator, but we argue that each intermediate node might alter the plaintext at will and affix its own signature to the modified version, if it only were not trusted not to do so: "Those parties are then trusted to enforce the privacy restrictions as specified by an individual" [17]. Removing this portion of trust would require each intermediate node to verify the customer's signature through a PKI.

Each flow may be seen as a delegation, as it is regulated by typical delegation mechanisms. Initially, the customer delegates a node to handle his data, and then the node delegates another one, and so on. The customer can block or unblock the use of his data at a node's through dedicated protocols, and the block message will propagate to all nodes that received the customer's data. More precisely, the block protocol sees the customer send an authenticated block request message to the first node in the data flow. If that node ever disclosed the data to others, then it now forwards the block request message, otherwise it responds to the customer with a signed block response message as an acknowledgement. Along each delegation path, the delegation response messages are nested. The unblock protocol is simpler: it sends the unblock request messages through the delegation graph but requires no response messages.

The WS protocol has the pro of being simple and intuitive. As the participating nodes are trusted to follow the protocol rules, the protocol only needs to regulate the flows of data among a distributed community. Therefore, an important remark is that no security mechanism to thwart the nodes' active misbehaviour is needed *within* the protocol because "our concept needs to be augmented by proper auditing and controls to ensure that enterprises correctly deploy the technology and comply with the privacy promises they have made" [17]. The same remark applies to the private information that each privacy panel features about a customer's data flows: access to that information should be forbidden to other customers.

3.2. Anonymity: Using a Pseudonym

Nothing in the previous paradigm confirms that the nodes will respect their stated privacy policy and therefore conform to the customer's policy upon reception of his data. Addressing this issue is left to other layers of technology. The opposite paradigm disposes with such a trust entirely by providing the customer with anonymity. Although pseudonymity and anonymity have different shades of meaning, our treatment safely makes no distinction between them — the interested reader may refer to other publications [18].

The DAA (Direct Anonymous Attestation) protocol [19, 20], which is adopted in the TPM v1.2 (Trusted Platform Module) specification by TCG (Trusted Computing Group), perhaps is the best-known protocol aiming at customer's anonymity. Here, we only outline in the context of electronic commerce the main steps of the protocol (Figure 3), as its details are irrelevant to the rest our treatment. The customer C is depicted paired with the TPM of his machine, which contains a unique endorsement key pair. Cryptographic operations may then be asked to the customer, as they will be performed by his TPM.

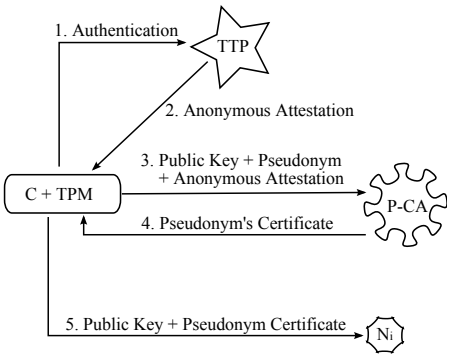

Figure 3: Using a pseudonym

An Issuer authenticates a customer through his pre-existing certificate, and issues an anonymous attestation (message) for him. This takes place through steps 1 and 2. The attestation message, which is encrypted with the customer's public key, states that the customer is genuine but does not reveal his identity. In

step 3, the customer generates a public key, chooses a pseudonym, and submits them to an entity called P-CA (Privacy Certification Authority). The P-CA verifies that the customer has valid attestation, and that his pseudonym is computed out of information that is present in the attestation. If this double check succeeds, then the P-CA will release the certificate for the pseudonym's public key, in step 4. Through this correctly attested purchase certificate, the customer can access the chosen end node N_i, taking step 5.

It can be appreciated that the protocol protects the customer's privacy, that is his identity and his TPM's endorsement public key, as these are only used in the initial phase with the Issuer. More precisely, for the attestation to be anonymous, an attacker must be unable to link the pseudonym with the customer's identity. Because the Issuer is the only entity that can resolve that link, the DAA protocol protects it by adopting a group of Issuers and a group signature scheme [21]. Moreover, the protocol separates the Issuer from the P-CA, so that a successful attack would require collusion with both authorities. The protocol can be additionally strengthened by having the P-CA release one-time certificates [20]. Some real-world applications adopting a DAA protocol already exist [22].

Also the concept of k-anonymity [23] is related to anonymity, although it does not rely on a pseudonym. It establishes that a tuple of records in a database is associated to k individuals so that the tuple cannot be uniquely associated to the actual data owner, who then remains anonymous. By contrast, we shall see below that data in our paradigm can be associated to their owner although, rather than being private data as such, they only are cover data.

4. A Novel Paradigm of Privacy Enforcement in E-Commerce

The previous Section presented two paradigmatic approaches to privacy enforcement. One requires the customers to trust the network, while the other one keeps the customers anonymous because that trust is removed. Our aim is to conjugate the benefits of both paradigms by removing as much as possible the trust from the network and by shifting the privacy enforcement mechanism from the customer's identity, which is a shift from anonymity, to the customer's actual required resources. In other words, the shift is from privacy of the customer's Personally Identifying Information (PII), which is "any piece of information which can potentially be used to uniquely identify, contact or locate a single person" [24], to privacy of the customer's information that is not PII, such as his purchase preferences. This Section assumes that the sought resource names do not include the customer's PII. Our paradigm will clearly avoid the necessity of a privacy certification authority. Of course, we cannot aim at removing the customer's trust from every node in his network: at least one node that will eventually provide the required resources or products must be trusted to keep the customer's privacy.

The main idea underlying our paradigm is to conceal a customer's data by means of a differential-privacy preserving function, and to transmit only the output of the function, that is cover data. It may be argued that also certain

cover data may expose private data. Precisely, the more probably cover data conceal private data, the less trust is required of peer nodes — for example, the customer may not want to share his interest in war books with the network because he does not trust its nodes sufficiently; conversely, he might accept to share that information with some probability he finds adequate because his trust in the network is proportionate to that probability. Balancing concealment of the input, that is privacy, with preciseness of the output, that is accuracy of the cover, can be done by resorting to differential privacy, as we shall see below (§5). The expected price to pay when trust tends to nothing is an increasingly inaccurate proposal finding, due to the fact that the cover data will match the original resource name with decreasing probability.

Our e-commerce privacy paradigm is a tradeoff between the customer's anonymity and his trust over the network. It comprises three phases.

1. **Data concealment** is the first and main phase as it operates the main shift of privacy enforcement from the customer's identity to his data. Concealment is done by applying a suitable differential-privacy preserving function (§4.1).
2. **Orchestration** then informs the customer of the network nodes providing the resources or the goods that best match his data. Obviously, a deeper data concealment causes a less precise orchestration (§4.2).
3. **Completion** sees the customer choose a node on the basis of its product offer. The customer finally initiates an appropriate security protocol (depending on the application domain) with the chosen node (§4.3).

As we shall see, all phases are as light-weight as possible for the customer. They can be easily implemented in a *privacy-preserving e-commerce service*, which, downloaded onto the customer's machine, would mechanise the interaction with the customer. He would only be left with the task of entering his required resources and wait for the best-matching offer.

4.1. Data Concealment

A customer who wishes to interact with a network to obtain specific resources begins by finding an "initial" node in that network. A deeper discussion about this search is beyond our interests here. For example, in the context of electronic purchases, that node might be accessed through the web site of a price-finding engine; in the context of peer-to-peer networks, the initial node has some "proximity" relation with the customer. For brevity, the initial node is termed *hook* in the following.

Phase 1 of our paradigm prescribes the customer to run SEP+ (§2.3) with the hook. Figure 4 portrays this phase from left to right. First, the customer C executes SEP+ with the hook N_1 inserting the baits in his required resource name. Then, the hook applies a differential-privacy preserving function f to the enriched resource name in order to produce cover data — more technicalities can be found below (§5). It is clear that using the original SEP protocol would be entirely inappropriate here, especially for a decentralized and delocalized

application as e-commerce. It cannot be assumed that any node anywhere in the world would collaborate to its own indictment without having digitally signed anything.

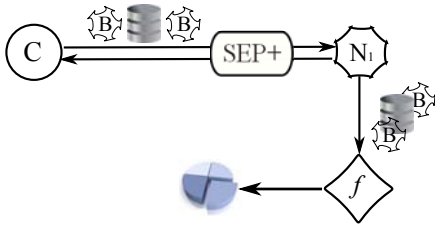

Figure 4: Phase 1 — data concealment

The hook is acting in the pollster's place. To operate in the individuals' role, the customer transmits the hook his required resource name modified with a number of baits. Our privacy-preserving e-commerce service can mechanise this interaction as the *data-concealment sub-service*, which takes as input a resource name and outputs its version enriched with the baits. Running such a sub-service seems a much lighter design requirement than having the customer compute the differential-privacy preserving function most appropriate to the type of shopping. SEP+ conveniently leaves that computation to the hook.

4.2. Orchestration

This phase begins when the hook has computed the cover data for the resources that the customer required. Figure 5 shows that the hook N_1 begins to transmit the cover to a number of participating nodes in the network (towards the right hand side of the Figure). It is important to remark that the customer's privacy is considered unaffected, according to the customer's privacy policy, because only cover data are treated. However, they can be linked to the customer's identity: he is not anonymous.

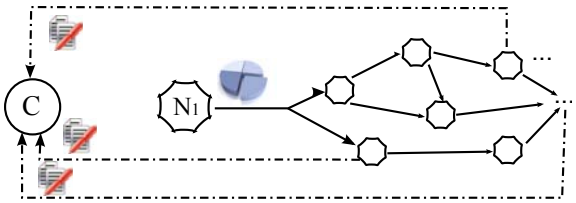

Figure 5: Phase 2 — orchestration

Transmission is recursive in the sense that whichever node receives the data will forward them to other nodes depending on its computational resources, anti-DoS heuristics and possibly a personal relationship that is either business-relevant (rational) or sentimental (irrational). In case of a business-relevant relationship, the node typically relies on some network reputation system. For

example, if a node N_i passes to a node N_j some data concerning stationery in the observed period of time, and stationary is relevant to N_j's business, then the reputation of N_i at N_j may increase; this may not be the case if the transmitted data concerned vehicles, upon the assumption that N_j does not sell them. Also, every node shall decrease all values in its reputation table proportionally with the passing of time, so as to promote the orchestration. However, maintaining a reliable reputation systems against potential false claims is not trivial [25] and lies beyond our focus.

When a node feels that it can make a significant offer about the data just received, it emails the customer with the details of the offer using a certified email protocol, as indicated by the dash-dotted lines in Figure 5. Alternatively, this process could be made more synchronous by suitable protocols [26]. Clearly, the level of accuracy of the offers depends on the cover data. The more privacy-preserving the cover, as established by the differential-privacy preserving function, the less focused the offers. Alternatively, the node may opt for a careful conduct during a valuable transaction: rather than sending the customer an offer that it recognizes as unappealing (that is either unrelated or overpriced), it may decide to pass the data on to other nodes with reputation higher than a threshold, and hence build up its own reputation. The node might even decide to do both but this will raise the competition.

In brief, a reputation system that accounts for the business-relevance of the transmitted data keeps the orchestration alive, because a simple cost-benefit analysis will convince each node to participate. In particular, each node may choose whether:

- to email an offer to the customer but not to pass on the data to other most reputed nodes, if the node cares more about the exclusiveness of its offer than about its network reputation;
- not to email an offer to the customer but to pass on the data to other most reputed nodes, if the node cares less about the exclusiveness of its offer than about its network reputation;
- to do both, if the node accepts the competition with other offers for the sake of increasing its network reputation.

We remark that how to make the best business choice among these three is not obvious for a node, as subsequent offers by other nodes might be higher as well as lower priced. In principle, a node might choose a greedy policy of always emailing offers but, due to the number of customers, processing all data containing customers' requests would raise the risk of a DoS attack to itself more than to a specific customer. It is however certain that the node would be unable to forge more expensive offers by other nodes because all offers must be made by certified email. This applies in particular to the hook, whose reputation gets balanced with its sole right of sale. The orchestration terminates with the customer's choice of the best offer. This will contain the very resources the customer is seeking, depending not only on their availability in the network but also on the accuracy of the cover data.

This phase does not significantly raise the risks of DoS attacks to the customer. The customer may decide to process the offers only for a limited time window, and to discard them afterwards. Moreover, he may only process lightweight emails containing the URL with the dedicated offer and discard the others. These heuristics would be simple to implement as the *orchestration sub-service* of our privacy-preserving e-commerce service to be run at the customer's machine. Ideally, the form of the offers should be standardised so that they could be mechanically selected: the orchestration sub-service would choose them if under a threshold price.

4.3. Completion

This final phase, which begins when the customer has already chosen the node from which to obtain his required resources, lets the customer come to a formal agreement with that end node. Figure 6 shows that the customer C executes a security protocol with the chosen end node N_i in order to settle a secure access to the resources. Obviously, the customer must reveal to the end node the required resources, but the security protocol shall protect its name.

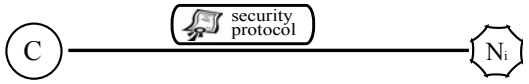

Figure 6: Phase 3 — completion

As remarked above, the customer must put some trust in the chosen end node. It is the only network entity trusted to conform to the customer's privacy policy, as opposed to the trust paradigm (§3.1) where the customer had to trust the entire network. Because the customer is not protected by anonymity, such a single-node trust cannot be removed. The end node will one way or another realise what resources to grant the customer or which good to ship to him. However, this limitation is somewhat shared also with the anonymity paradigm (§3.2).

Security in this phase strongly depends on the application domain. It may generically evaluate to mutual authentication and confidentiality. For electronic commerce in particular, this phase would require a suitable protocol such as SSL/TLS [3] or a fuller payment protocol such as SET [27]. In addition, a combination with a fair-exchange protocol [28] would protect the peers from each other's potential false claims. This would help the customer face the risks associated to his trust on the end node. The protocol would require the end node to formally agree with the customer's privacy policy, so that such agreement would be non-repudiable. Should the end node abuse the customer's private data, that is what he has just purchased, the customer would be able to sue it using the non-repudiation evidence collected during the protocol.

The interaction with the customer in this phase reduces to an appeal to an appropriate security protocol. The *completion sub-service* of our privacy-preserving e-commerce service would only have to implement that invocation.

The implementation effort would be negligible: SSL for example is implemented in all modern browsers, while open-source implementations are already available for it [29].

5. Simulating data concealment with a candidate differential-privacy preserving function

Given two alphabetic words $X = x_1 \dots x_n$ and $Y = y_1 \dots y_n$ of same length $l(X) = l(Y) = n$, we define their *similarity* as the number of symbols they have in common, in specific positions, as:

$$s(X,Y) = \sum_{i=1}^{n} j \ : \ j = \begin{cases} 1 & \text{if } x_i = y_i \\ 0 & \text{else} \end{cases}$$

For example, $s(\text{privacy}, \text{wrivxcg}) = 4$. To introduce a differential-privacy preserving function, it is useful to compare and contrast it with an analogous deterministic function.

Let $\mathcal{D}$ denote the set of all English words; it is larger than the English dictionary because it includes all dictionary entry variations such as plural nouns and conjugated verb forms. A deterministic extractor of the most similar word to a given one can be defined as:

$$e_{det}(X) = Y \ : \ Y \textit{ is alphabetically the first in } \{W : W \in \mathcal{D} \ \wedge \ l(W) = l(X) \ \wedge \ s(X,W) = \max_{P \in \mathcal{D}} s(X,P)\}$$

The function begins by building the subset of dictionary words of the same length as that of X, and such that they bear maximum similarity to X. It terminates by deterministically choosing a word in that subset, for example the first in alphabetical order.

Then, an analogous though differential-privacy preserving extractor can be defined resorting to a probability distribution over the neperian number as:

$$e_{dpp}(X) = Y \textit{ with probability proportional to } \mathrm{e}^{\,\epsilon\, s(X,Y)}$$

According to an established result [10, Theorem 6], this function is 2ϵ-differential-privacy preserving because it is built using the so called exponential mechanism. Further details lie outside our focus, but it must be stressed that the probability with which the function provides a specific output Y is not exactly $\mathrm{e}^{\,\epsilon\, s(X,Y)}$ but, rather, proportional to it. Therefore, the exact probability can be obtained by normalising the probability pre-distribution $\mathrm{e}^{\,\epsilon\, s(X,Y)}$.

To get to grasps with e_{dpp}, it is useful to observe how the probability pre-distribution varies with its exponent. In particular, Figure 7 shows how the probability pre-distribution varies with the similarity function, having set ϵ increasingly smaller, first as 0.1, then as 0.01, and finally as 0.001. It can be appreciated that, as ϵ gets smaller, the pre-distribution tends to flatten irrespectively of the similarity function. The consequences are significant, as we shall see.

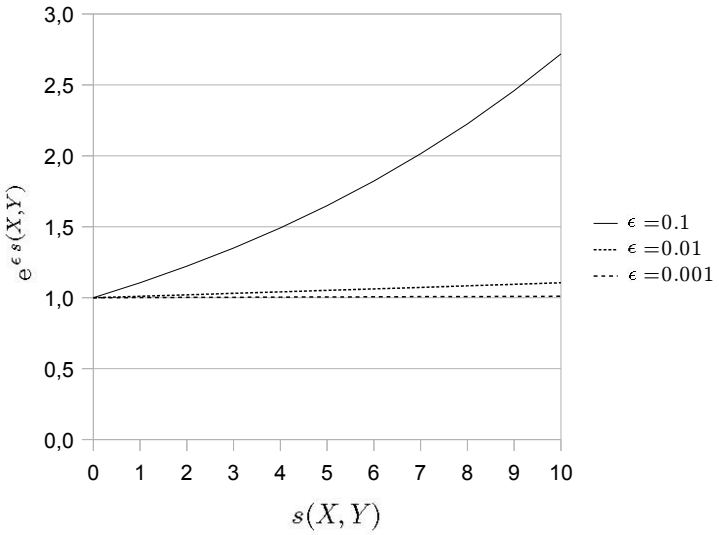

Figure 7: The similarity function in abscissas and the probability pre-distribution in ordinates, having fixed ϵ to 0.1, 0.01 and 0.001 respectively

We have implemented a Java prototype simulator for the data concealment phase in approximately 200 lines of code. Therefore, to demonstrate this phase to the reader, we discuss some of the findings obtained with the help of the prototype. Each letter is represented by the five least significant bits of the ASCII binary code so that all codes can be printed; for example "a" is represented as 00001 and "z" as 11010. We premise that our interpretation of the insertion of a bait is that the bait replaces a bit in a specific position. Because the number of baits does not influence the chances of indictment [4], we decide to insert a number of baits that is 20% of the total length of the binary representation, precisely one bait per quintuple. So, each letter gets a bait, but because the baits are randomly generated, this does not imply that all letters will change.

Let us assume that the customer seeks a resource named "privacy". We run our simulator with the bait insertion parameters set above, taking as $\mathcal{D}$ a wordlist of approximately 5MB [30]. At the first run, we obtain output "wrivxcg" as the word that the hook decrypts through SEP+. This word still has the four letters "rivc" matching the original resource name in specific positions. Because of the randomness of the baits, at the second run we obtain output "airways". Our simulator also finds out that "wrivxcg" has 2677 words with similarity 1, 447 words with similarity 2, 46 with similarity 3, and 4 with similarity 4, but no other words with higher similarity. By contrast, because "airways" belongs to the wordlist, the simulator obviously finds it with maximum similarity, that is 7. Table 1 summarises these findings. Among other information, it can be seen that "airways" only has 2 words with similarity 5 and no words with similarity 6. Also, the Table shows that "privacy" has

similarity 4 to the first word and 1 to the second.

X	wrivxcg	airways	$s(X,Y)$
Y	abasing abating ability absence ... *(2677 words)*	abalone abandon abashed abasing ... privacy ... *(4486 words)*	1
	abiding alining arching arking ... *(447 words)*	abashes abducts abjures ablates ... *(1307 words)*	2
	arising braving bribing craving ... *(46 words)*	abrades abreast acreage acronym ... *(160 words)*	3
	driving privacy waiving writing	areials affrays airbase airfare ... *(tot. 22 words)*	4
		airbags midways	5
			6
		airways	7

Table 1: Sample of English words w.r.t. their similarity to "wrivxcg" and to "airways"

It follows that $e_{det}(\text{wrivxcg}) = \text{driving}$, as "privacy" is not alphabetically the first in its class, and that $e_{det}(\text{airways}) = \text{airways}$. By contrast, the homologous observation about e_{dpp} is that:

$$\Pr[e_{dpp}(\text{wrivxcg}) = \text{privacy}] \;=\; \Pr[e_{dpp}(\text{airways}) = \text{airbase}]$$

because

$$S(\text{wrivxcg}, \text{privacy}) = S(\text{airways}, \text{airbase})$$

However, observations of this kind are not relevant to our purposes. A first useful observation is that, having fixed the ϵ parameter, the probability

distribution with which e_{dpp} outputs a word is identical from column to column in Table 1 because it only depends on the similarity.

What really is significant is to observe how little the probability distribution varies within each column: e_{dpp} may yield "privacy" almost equally as it may yield the other words in the column because, for sufficiently small values of the ϵ parameter such as 0.001, the probability pre-distribution is almost flat, irrespectively of the similarity to the input word.

More precisely, e_{dpp}(wrivxcg) may return "privacy" with a probability that is only negligibly higher than the probability with which it may return words, such as "ability", with lower similarity to the input. Figure 7 confirms this: the probability pre-distribution varies negligibly, from 1.001 to 1.004, when the similarity changes from 1 to 4. Likewise, e_{dpp}(airways) may return "privacy", despite the low similarity of the two words, with probability that is only negligibly lower than that with which it may return "airbase", while arguably a reasonable deterministic extractor would not output "privacy" in this case.

In conclusion, our ϵ-differential-privacy preserving extractor can be used during the data concealment phase to compute cover data because it exhibits the desirable properties of outputting:

- rather equiprobable words irrespectively of their similarity to the input word when ϵ is relatively small, and
- discernibly probable words, depending on their similarity to the input word, when ϵ is relatively big.

In line with ϵ-differential privacy [4], this means that our extractor can be tuned towards concealment of the input, that is privacy, or towards preciseness of the output, that is accuracy, by appropriately setting the ϵ parameter.

Given an input word of length l, the words of length l in $\mathcal{D}$ that our extractor may yield can safely be assumed equiprobable for adequately small values of ϵ. Therefore, the probability of an attacker to invert the output of our extractor and find the input word is inversely proportional to the cardinality of the subset of $\mathcal{D}$ of length l. Such cardinality can be assumed to be bigger than the cardinality of the possible domain of answers to e-polls. Hence, our application of differential privacy seems more robust in terms of privacy than the application to e-polling.

To the best of our knowledge, this is the first practical definition, demonstration and application of a differential-privacy preserving function with non-numeric values. An obvious limitation is that it becomes increasingly difficult to conceal the input while its length increases. This is due to the fact that there exist few very long words. We are currently working towards more intelligent extractors that break up long words suitably. It is already clear that, working with non-numeric values, the main limitations that our definitions must face live in the world of linguistics.

6. Discussion

The authors of the WS protocol remark that "without additional assumptions it is impossible to achieve the correct schema against a node that is completely untrusted" [17]. Their protocol cannot work without enforcing trust in the network at another architectural level. For example, nothing in the protocol prevents an intermediate node to tamper with the ACLs and the DF matrix that come with the customer's data, thus violating the customers' privacy. The ACLs and DF matrix are digitally signed by the customer but the node is not prescribed to verify those signatures. Even so, the node might replace the data signature with his own, or pretend to have never received the data or sell them before actually accepting a block request. The customer would have no means to prove its misbehaviour.

We have seen that our privacy paradigm transmits the customer's data only to the hook. The raw names of the resources the customer is looking for can be enriched with baits because the hook will have to compute cover data out of them, as prescribed by SEP+, otherwise it could be indicted. Therefore, it is unnecessary to program a privacy panel into all nodes' web servers. Also, the ACLs and the DF matrix are not required because cover data can flow freely between the nodes. The orchestration is guaranteed to proceed by the cost-benefit analyses of the nodes, while each node will freely decide how to behave (whether to make an offer and/or to forward the cover data) according to its own business logic.

Our paradigm pays a price when cover data that fail to match the required resource name are orchestrated. The hook can be required to launch the orchestration repeatedly, each time with fresh cover data, for a fixed number of times inversely proportional to the ϵ parameter. More offers would be generated, but the orchestration sub-service could support the customer in mechanically choosing among variously pertinent offers. Should the customer remain dissatisfied with all offers after a fixed number of orchestrations, then he could protect his privacy by restarting the data concealment phase with the same resource name: the randomness of the baits would leave the hook unaware.

All three paradigms allow a man-in-the-middle attack that sees the hook behave as the attacker. Nothing can prevent it from restarting the entire protocol abusing the customer's data as if they were its own. Assuming that the subsequent orchestration phase would not make the data public, the hook would not be indictable because the customer would remain unaware. The hook might finally collect all offers and abuse them analogously with the customer, that is as offers of its own. In case of resources for sale, for example, the hook might present increased prices to the customer. It seems impossible to prevent this scenario even in the real world, where a shop might sell the same products of another shop though at an increased price, while the clients ignore it. Anyone who can both provide and seek resources becomes a potential attacker. However, our paradigm at least reduces the chances of a successful attack because an offer will not necessarily match the actual resources that the customer was looking for. This attack is limited in practice by the rules of the open market,

which see various sub-markets compete for the best prices.

A simple man-in-the-middle attack would have threatened the orchestration phase of our privacy paradigm if the offers were not submitted by certified email, which makes the email contents confidential for its peers. The attack would have seen a node intercept and block an offer being emailed, and then make the same offer as one of its own. The price of the fake offer could have been higher or lower than the original offer's depending on whether the original could be blocked or not.

7. Conclusions

We have advanced a novel paradigm to safeguard customers' private data in e-commerce. It is the natural balance between the paradigm of trusting the network, adopted by the WS protocol, and the paradigm of distrusting it entirely thanks to anonymity, adopted by the DAA protocol. Trust is reduced to the single node selling the required resource to the customer. Privacy of identity, that is anonymity, is shifted towards privacy of data. More precisely, the original data are concealed and therefore both the need for full network trust and for anonymity are removed.

The thorough analysis of the SEP protocol for e-polls has denounced lack of fairness for the participants. We have upgraded the protocol as SEP+ and adopted it in the data concealment phase of our privacy paradigm, so that the customer can safely leave the computation of cover data to someone else. The orchestration phase delivers pertinent offers to the customer by certified email. The completion phase requires the execution of appropriate security protocols between the customer and the chosen selling node. Customers should run a privacy-preserving e-commerce service that mechanises such paradigm.

Moreover, we have defined and demonstrated what seems to be the first differential-privacy preserving function with non-numeric values. With the help of our Java prototype simulator, we have shown how to apply the new function profitably in the data concealment phase.

E-commerce is developing quickly, especially with the advent of the Web2.0, as with eBay for example. The inherently hierarchical architecture of the Internet is flattened at the application level so that each node may easily become both a customer and a merchant. This new picture makes trusting the network difficult to accept, and the frequent appeal to a trustworthy privacy certification authority perhaps more problematic. Our privacy-preserving e-commerce paradigm may then offer valid support to the new picture.

Acknowledgments.. This work was partially supported by the FP7-ICT-2007-1 Project no. 216471, "AVANTSSAR: Automated Validation of Trust and Security of Service-oriented Architectures" (`www.avantssar.eu`). Francesco Librizzi contributed to an earlier version of our privacy paradigm. Matthias Schunter kindly provided extra clarifications about the WS protocol. Giovanni Russo helped with the probability pre-distribution. The anonymous reviewers' comments were of great help to improve the presentation.

References

[1] K. J. Hole, T. Tjøstheim, V. Moen, Next generation Internet banking in Norway, Tech. rep., NoWires Research Group (2008).

[2] B. C. Neuman, T. Ts'o, Kerberos: An authentication service for computer networks, IEEE Communications Magazine 32 (9) (1994) 33–38.

[3] T. Dierks, C. Allen, The TLS Protocol, Internet Request for Comment RFC-2246 (January 1999).

[4] P. Golle, F. McSherry, I. Mironov, Data collection with self-enforcing privacy, ACM Transactions on Information and System Security 12 (9) (2008) 1–24.

[5] G. Bella, F. Librizzi, S. Riccobene, Realistic threats to self-enforcing privacy, in: M. Rak, A. Abraham, V. Casola (Eds.), Proc. of the 4th International Symposium on Information Assurance and Security (IAS'08), IEEE Press, 2008, pp. 155–160.

[6] G. Bella, F. Librizzi, S. Riccobene, A privacy paradigm that tradeoffs anonymity and trust, in: N. Rozic, D. Begusic (Eds.), Proc. of the 2008 International Conference on Software, Telecommunications and Computer Networks (SoftCOM'08), IEEE Press, 2008, pp. 384–388.

[7] M. Stegelmann, Towards fair indictment for data collection with self-enforcing privacy, in: K. Rannenberg, V. Varadharajan, C. Weber (Eds.), Security and Privacy Silver Linings in the Cloud, Vol. 330 of IFIP Advances in Information and Communication Technology, Springer, 2010, pp. 265–276.

[8] C. Dwork, Differential privacy, in: M. Bugliesi, B. Preneel, V. Sassone, I. Wegener (Eds.), Proc. of 33rd International Colloquium on Automata, Languages and Programming (ICALP'06), LNCS 4052, Springer, 2006, pp. 1–12.

[9] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: S. Halevi, T. Rabin (Eds.), Proc. of 3rd Theory of Cryptography Conference (TCC'06), LNCS 3876, Springer, 2006, pp. 265–284.

[10] F. McSherry, K. Talwar, Mechanism design via differential privacy, in: Proc. of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07), IEEE Press, 2007, pp. 94–103.

[11] G. Bella, Formal Correctness of Security Protocols, Information Security and Cryptography, Springer, 2007.

[12] G. Bella, S. Bistarelli, F. Massacci, Retaliation: Can we live with flaws?, in: M. Essaidi, J. Thomas (Eds.), Proc. of the Nato Advanced Research Workshop on Information Security Assurance and Security, Volume 6 of NATO Security through Science Series, IOS Press, 2005, pp. 3–14.

[13] D. Dolev, A. Yao, On the security of public-key protocols, IEEE Transactions on Information Theory 2 (29) (1983) 198–208.

[14] W. Arsac, G. Bella, X. Chantry, L. Compagna, Validating security protocols under the General Attacker, in: P. Degano, L. Viganò (Eds.), Proc. of the Joint Workshop on Automated Reasoning for Security Protocol Analysis and Issues in the Theory of Security (ARSPA-WITS'09), LNCS 5511, Springer, 2009, pp. 34–51.

[15] G. Bella, L. C. Paulson, Accountability protocols: Formalized and verified, ACM Transactions on Information and System Security 9 (2) (2006) 1–24.

[16] R. Cramer, V. Shoup, Signature schemes based on the strong RSA assumption, ACM Transactions on Information and System Security 3 (2000) 161–185.

[17] M. Schunter, M. Waidner, Simplified privacy controls for aggregated sevices — suspend and resume of personal data, in: N. Borisov, P. Golle (Eds.), Proc. of 7th International Symposium on Privacy Enhancing Technologies (PET'07), LNCS 4776, Springer, 2007, pp. 218–232.

[18] A. Pfitzmann, M. Hansen, A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, http://dud.inf.tu-dresden.de/Anon_Terminology.shtml (2010).

[19] E. Brickell, J. Camenisch, L. Chen, Direct anonymous attestation, in: Proc. of the 11th ACM conference on Computer and communications security (ACM CCS'04), 2004, pp. 132–145.

[20] J. Camenisch, Better privacy for trusted computing platforms, in: P. Samarati, P. Y. A. Ryan, D. Gollmann, R. Molva (Eds.), Proc. of the 9th European Symposium on Research in Computer Security (ESORICS'04), Vol. LNCS 3193, Springer, 2004, pp. 73–88.

[21] G. Ateniese, J. Camenisch, M. Joye, G.Tsudick, A practical and provably secure coalition-resistant group signature scheme, in: Proc. of the 20th International Conference on Advances in Cryptology (CRYPTO'00), LNCS 1880, Springer, 2000, pp. 255–270.

[22] E. Gallery, C. J. Mitchell, Trusted mobile platforms, in: A. Aldini, R. Gorrieri (Eds.), Tutorial Lectures on Foundations of Security Analysis and Design IV (FOSAD'06/07), LNCS 4677, Springer, 2007, pp. 282–323.

[23] L. Sweeney, k-anonymity: a model for protecting privacy, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10 (2002) 557–570.

[24] Wikipedia, Personally Identifiable Information (PII), http://en.wikipedia.org/wiki/Personally_identifiable_information (2010).

[25] K. Hoffman, D. Zage, C. Nita-Rotaru, A survey of attack and defense techniques for reputation systems, ACM Computing Surveys 42 (2009) 1–31.

[26] P. Riikonen, SILC protocol white paper, http://silcnet.org/docs/silc_protocol.pdf (2003).

[27] G. Bella, F. Massacci, L. C. Paulson, Verifying the SET purchase protocols, Journal of Automated Reasoning 36 (1-2) (2006) 5–37.

[28] J. Zhou, D. Gollmann, A fair non-repudiation protocol, in: Proc. of the 15th IEEE Symposium on Security and Privacy (SSP'96), IEEE Press, 1996, pp. 55–61.

[29] The OpenSSL Project, OpenSSL: The open source toolkit for SSL/TLS, http://www.openssl.org/.

[30] K. Atkinson, Official 12dicts package, http://wordlist.sourceforge.net/ (2011).