PROTECTING CONSUMER DATA IN COMPOSITE WEB SERVICES

Craig Pearce, Peter Bertok, Ron Van Schyndel
RMIT University. Melbourne, Australia
{crpearce,pbertok,ronvs}@cs.rmit.edu.au

Abstract: The increasing number of linkable vendor-operated databases present unique threats to customer privacy and security intrusions, as personal information communicated in online transactions can be misused by the vendor. Existing privacy enhancing technologies fail in the event of a vendor operating against their stated privacy policy, leading to loss of customer privacy and security. Anonymity may not be applicable when transactions require identification of participants. We propose a service-oriented technically enforceable system that preserves privacy and security for customers transacting with untrusted online vendors. The system extends to support protection of customer privacy when multiple vendors interact in composite web services. A *semi-trusted processor* is introduced for safe execution of sensitive customer information in a protected environment and provides accountability in the case of disputed transactions.

Key words: Electronic commerce; privacy; security; web services.

1. INTRODUCTION

Many vendors have shown poor security of customer databases, leading to intrusions, loss of customer privacy and even identity theft [internetnews.com, 2003].

When back-end customer databases are copied, sold or linked with databases of other vendors, the wealth of available customer information rapidly increases. In some cases, customers trust a vendor with personal information, however the information is collected for processing by other (untrusted) parties along the chain, as seen in outsourcing and supply chain management [Medjahed et al., 2003].

Currently, private information that customers choose to release to vendors, such as medical information or credit card details, cannot be fully controlled by the customer once released. In addressing this issue, we have designed a generalised application-layer privacy platform, named: TEPS, the Technically Enforceable Privacy and Security system. TEPS protects from customer privacy violations at the vendor-side by preventing an untrusted vendor from ever holding customer personally identifiable information (PII) in plain view. The customer decides which of their personal attributes to protect and we introduce a semi-trusted processor (STP) that is trusted not to disclose customer PII within local execution of vendor-provided business logic. Full trust of the STP is not required as accountability and code watermarking [Collberg and Thomborson, 2002] can detect other forms of STP abuse. Mobile code is utilised as a method of communicating messages of varying protection levels amongst the entities of the service-oriented electronic commerce architecture.

TEPS is a generalised model, and is suitable within the Web Services architecture, where multiple vendors can interact to fulfill customer requests, typically seen with a front-end web service broker that outsources back-end activities to other web services.

Our results from a fully scaled implementation within wired and wireless networks, and the possibility of mobile clients, show that TEPS is suitable within service-oriented transactions, enforcing consumer privacy as a value-added service.

2. BACKGROUND AND RELATED WORK

Traditionally, once a vendor has access to plain-text (non-encrypted) customer information, there are no technical methods available to restrict its use of that information.

Anonymising layers, such as [Chaum, 1981, Jakobsson and Juels, 2001, Dingledine et al., 2004], help protect the customer source identity, and sometimes vendor destination, but once personally identifiable information has been captured by the vendor it can no longer be controlled. Identity Management systems, such as [Waldman et al., 2000, Campbell et al., 2002, Jendricke et al., 2004], act as an intermediary between customer and vendor and provide a pseudonym of the customer instead of the customer's real identity. This establishes privacy as long as pseudonyms cannot be linked to the customer's real identity. However, pseudonyms cannot be used when a vendor is required to authenticate a customer in environments that provide services both in electronic and traditional environments, such as banking, voting and payment. Credential-carrying pseudonyms [EU FP6 PRIME Project, 2005] could be considered an alternative to strong authentication, but require globally present identity management mechanisms.

Non-traceable anonymous payment systems, such as [Chaum, 1982, Chaum et al., 1990] for transactions requiring authentication remain to be problems, such as medical subscriptions and large order requests.

The Secure Electronic Transactions (SET) protocol used hashing techniques to preserve privacy of payment and order information, although overheads of client-side certificates, implementation difficulties and lack of extensibility for multiple vendors within integrated transactions made it unsuitable for complex environments, such as Web Services [Medjahed et al., 2003].

The Secure Sockets Layer (SSL/TLS) [Dierks and Rescorla, 2004] provides communication channel authentication, message confidentiality and integrity but protects only the communication channel between customer and vendor. Customer privacy from untrusted vendors is not protected once data has reached the vendor.

Protection of a customer's personally identifiable information (PII) has been proposed [Kenny and Korba, 2002] but does not offer assurance of enforceability in global e-commerce. Furthermore, the proposed PII-protecting model [Kenny and Korba, 2002] requires full trust in the data controller, which is also responsible for accountability. Personnel are required to manually check data processing activity and the security of data controllers is simplified to a question of reputation. Extensible support for multiple vendors interacting within a transaction has not been addressed.

Encrypting digital identifiers and enforcing associated privacy policies through trusted computing technologies [Casassa et al, 2003] has been suggested, however all participants are required to operate within the confines of a globally unified trusted computing platform.

Recent developments in XML-based privacy between customer and vendor has seen the emergence of Platform for Privacy Preferences (P3P)

[W3C, 2002, Berthold and Köhntopp, 2001] for the Internet and Enterprise Privacy Authorization Language (EPAL) [Ashley et al., 2003] for organisations. P3P and EPAL provide a standardised way for the vendor to represent their privacy policy and allow the customer to specify their privacy needs but cannot provide technical assurance that the vendor will not digress from their stated privacy policy. EPAL provides logging and reporting capabilities and enforces privacy access within an organization [Goldberg, 2002] using network privacy monitors, however, is not appropriate for complex transactions as customers are required to unconditionally trust resources governed by vendor organisations. Furthermore, P3P and EPAL were designed for web-based applications, using the traditional client-server model, and are not suitable for Web Services [Medjahed et al., 2003].

Issues of vendors digressing from their stated privacy policy, lack of identification and non-repudiation in anonymous payment systems, overheads of client identity certificates and legal factors due to globalisation have encumbered electronic commerce with privacy concerns. In many jurisdictions, revelation of customer databases to third parties is legally punishable if detected, but is still prevalent due to limitations in tracking down the perpetrator. Globalisation increases this problem as privacy laws in some jurisdictions are weak or non-existent.

The "Technically Enforceable Privacy and Security" (TEPS) system helps solve these core issues by operating as a generalised service at the application-level protocol layer, and is suitable in a service-oriented architecture to prevent vendors from ever gaining access to customer privacy information.

3. SCENARIOS: HOW ONLINE TRANSACTIONS AFFECT CUSTOMER PRIVACY

In this section we describe two realistic scenarios currently threatening customer privacy that TEPS aims to alleviate.

3.1 Scenario 1: Online brokers

A customer uses on online bookseller web service as the vendor to locate a textbook. After finding a suitable match, the customer decides to purchase the package from the vendor. Current practices require customers to log into the vendor's website with a previously established account that probes for customer identity information. SSL/TLS is used for encrypting credit card information, which is generally handled by a payment gateway, not the vendor. The vendor redirects customers to a payment gateway, and once payment is complete, the payment gateway returns an outcome to the vendor. Despite what may be stated within the vendor's privacy policy, SSL/TLS does not prevent the vendor from disclosing consumer spending habits to other parties.

3.2 Scenario 2: Composite web services

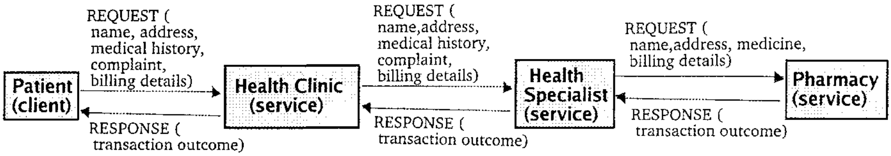

REQUEST (name, address, medical history, complaint, billing details)
Patient (client)
RESPONSE (transaction outcome)
Health Clinic (service)
REQUEST (name,address, medical history, complaint, billing details)
RESPONSE (transaction outcome)
Health Specialist (service)
REQUEST (name,address, medicine, billing details)
RESPONSE (transaction outcome)
Pharmacy (service)

Figure 1. Composite web services

A customer seeks medication by lodging a request to an online health clinic and must log in for identification. As with Scenario 1, the previously established account may require a number of personally identifiable customer attributes deemed private in nature. The health clinic is a front-end only, outsourcing medical knowledge to a specialist back-end service, as shown in Figure 1. Furthermore, if medicine is required, the specialist outsources prescription services to a pharmacy. The customer may not be aware of multiple vendors operating to fulfil their transaction. Each of these back-end services will request customer details from the front-end service to perform their business activity, possibly without customer knowledge. Privacy policies of back-end services may be independent to the health clinic privacy policy agreed to by the customer.

4. SYSTEM DESIGN

TEPS is composed of the following entities:

- **Customer (CUS):** Operates a client (CL) machine through a web browser;
- **Client (CL):** Computer used by customer in transacting with a vendor;
- **Vendor (V):** Service-oriented online store (for example, travel agent, weather service);
- **Semi-Trusted Processor (STP):** Partially trusted intermediary between client and vendor in processing vendor business logic on

customer PII data. Example STPs include payment gateways, identity verifiers and marketing bureaus to name a few;

- **Certificate Authority (CA):** Trusted certificate server used for distribution and revocation of digital certificates to the entities communicating in an online transaction. The CA can be used throughout online transactions for verification of certificates with public key encryption and signing;
- **Accountability Authority (AA):** Used in disputed transactions to provide accountability of participants in case of abuse. The *AA* stores hashes of information used within a transaction, saving space and providing confidentiality to the other parties. A transaction is disputed when enough threshold certificates are gathered from disputing parties or if requested by an external certified entity.

The *AA* and *CA* are essential services for a technically-enforceable system that guarantees privacy and accountability. The current approach to online transactions (Section 3, Scenario 1) uses SSL/TLS encryption and X.509 Certificates signed by certificate authorities (CAs) to communicate vendor certificates to clients. An accountability service is not provided, limiting the types of transactions performed online due to lack of defined dispute resolution mechanisms.

4.1 Assumptions

In formulating our system, we considered the following assumptions:

- *STPs* will not knowingly reveal PII data to another entity (with the exception of an accountability authority in pre-defined legal circumstances);
- *STP, AA* and Certificate Authority (*CA*) services are who they claim to be; host security has not been breached;
- Vendors comply with the privacy system by programming their business logic in a way that is executable by the *STP*;

These assumptions show the proposed solution to be useful in providing customer privacy protection in scenarios where vendors are willing to program and communicate their business logic to STPs. This is not a major overhead, as vendor business logic should be a direct implementation of the action stated publicly in their privacy policy. In cases of rigid intellectual property agreements, non-disclosure agreements (NDAs) or outsourcing could be negotiated between vendor and semi-trusted processor.

Additional privacy requirements, such as data minimisation and purpose binding can be met by the customer proactively reading the vendor's privacy policy and discontinuing the transaction if the collection purpose or amount of requested information is not appropriate.

We plan for TEPS to utilise existing privacy and security services where possible. While TEPS is a generalised model, this paper explores TEPS in a service-oriented environment, with Figure 2 showing the communication stack layering TEPS on top of web services, as web services alone do not protect customers from misbehaving vendors. SSL/TLS can be used for underlying channel communication security.

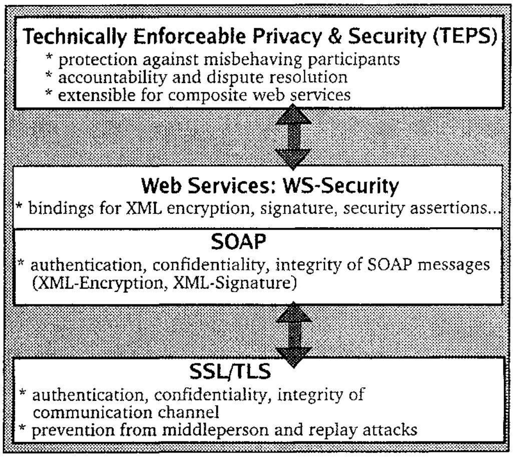

Technically Enforceable Privacy & Security (TEPS)
* protection against misbehaving participants
* accountability and dispute resolution
* extensible for composite web services
Web Services: WS-Security
* bindings for XML encryption, signature, security assertions...
SOAP
* authentication, confidentiality, integrity of SOAP messages (XML-Encryption, XML-Signature)
SSL/TLS
* authentication, confidentiality, integrity of communication channel
* prevention from middleperson and replay attacks

Figure 2. TEPS communication stack: privacy and security for Web Services

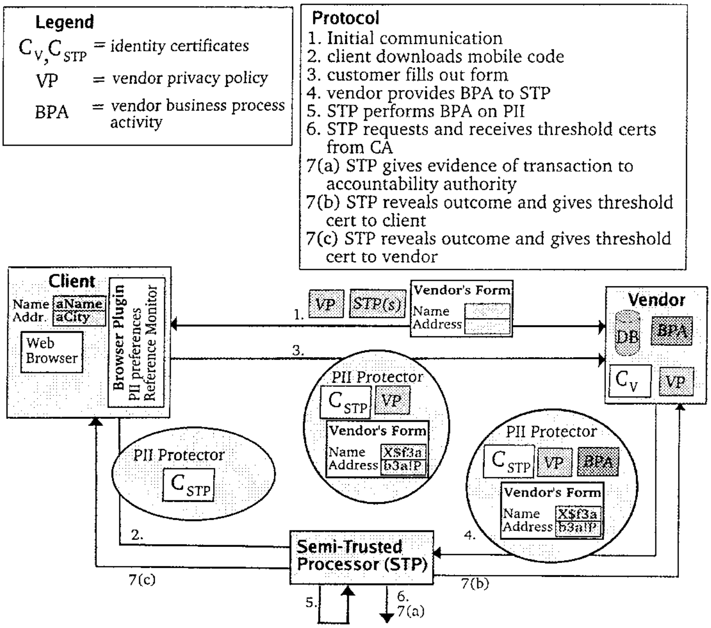
4.2 Processing of an online transaction

Figure 3 shows the functional steps taken in a transaction using TEPS. Each phase within Figure 3 is described here:

1. Whenever a vendor's form requests an input that has been marked PII, the client privacy reference monitor will transparently request a list of *STPs* from vendor. The vendor will compile a list of *STPs* (consulting a business registry *(BR)* if needed) and return this to the client with vendor's privacy policy *(VP)*. The client hashes the *VP* and stores it safely in case of a disputed transaction;
2. From a given a list of *STPs*, the client will choose one, and then contact it to download the PII protector mobile code, providing a name certificate for transfer of a temporal public key. The *STP* generates $K_{STP\text{-}CL}$, a shared secret key, encrypting it with the client's public key for confidentiality. The PII-protecting mobile code is signed by the *STP*;

3. The customer fills out the vendor's HTML form. The client executes *STP's* mobile code which protects the customer's PII by encrypting it with $K_{STP\text{-}CL}$;
4. Upon receiving the PII-Protecting mobile code, the vendor executes the mobile code which prompts for a business process activity (*BPA*);
5. Once the mobile code cycles back to the STP, the *BPA* is processed with customer's PII data in a safe environment;
6. Threshold certificates are provided by *CA* after providing the name certificates of participants in the transaction

7.*STP* communicates *h(VP), h(BPA),* $h(\{PII\}K_{STP\text{-}CL})$ hashes to *AA*. *STP* then responds to the vendor and client with the transaction outcome and threshold certificates in case a dispute arises;



Legend

C_V, C_{STP} = identity certificates

VP = vendor privacy policy

BPA = vendor business process activity

Protocol

1. Initial communication
2. client downloads mobile code
3. customer fills out form
4. vendor provides BPA to STP
5. STP performs BPA on PII
6. STP requests and receives threshold certs from CA

7(a) STP gives evidence of transaction to accountability authority

7(b) STP reveals outcome and gives threshold cert to client

7(c) STP reveals outcome and gives threshold cert to vendor

Figure 3. Privacy in transactions

The transaction will be aborted if the client is not satisfied with the list of *STPs* provided by the vendor in Figure 3 Step 1. If a party stops responding during the processing of a transaction, the transaction will time out and be aborted.

4.3 Composite web services

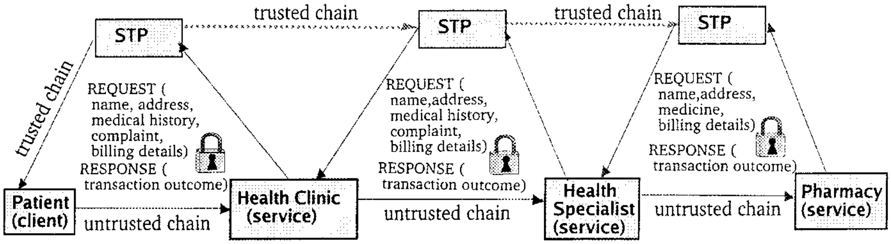


Figure 4. Technically enforced privacy and security in composite web services

Scenario 2 of Section 3 described a transaction involving a customer and multiple vendors. Web Services privacy is an open question when each vendor performs a separate business process activity, integrated to form a composite web service [Medjahed et al., 2003]. We address this issue by forcing the front-end web service to clearly state the need of back-end vendors in their privacy policy, and the client agreeing to transitivity of semi-trusted processing of personal information. The TEPS protocol is then performed recursively for each back-end vendor. For instance, the example composite web service in Figure 1 involves a separate invocation of TEPS for the Health Specialist and Pharmacy services, as shown here in Figure 4. Each subsequent vendor has an associated, possibly different, semi-trusted processor to perform its business process activity, preserving privacy for the previous vendor. A tree-based structure is formed and includes two chains of information flow: (1) untrusted vendor chain which has no access to client personally identifiable information or adjacent vendor privacy information and (2) trusted chain for semi-trusted processors to communicate customer personally identifiable information (PII) from top STP to bottom STP. While trust management of the STP chain is not addressed here, we assume clients to explicitly agree to adjacent STPs in a chain exchanging privacy information between themselves (transitivity).

4.4 Accountability and disputed transactions

Transactions may be disputed when two or more parties out of three submit a dispute request with their allocated threshold certificate.

Alternatively, external certified entities *(ECEs)* can initialise a disputed transaction by submitting a signed request with appropriate certification. An example scenario for *ECE* involvement would be law enforcement officers with reason to believe one of the parties committed fraud.

Possible disputed transactions include:

1. (*CL* AND *STP*) AGAINST *V*
2. (*V* AND *STP*) AGAINST *CL*
3. (*CL* AND *V*) AGAINST *STP*
4. *ECE* AGAINST (*STP* OR *CL* OR *V*)

Each party gives their evidence to *AA* who contains enough information to judge whether the defendant, first claimant and/or second claimant are cheating.

If the defending party is not contactable for any reason, the transaction is logged as 'in dispute' by *AA* and claimants.

The dispute resolution mechanism is a two-step protocol , with the AA firstly attempting to reach an outcome without knowledge of the PII-protecting key, $K_{STP\text{-}CL}$. If an outcome cannot be determined at this point, only then will the *AA* request submission of $K_{STP\text{-}CL}$ as evidence; both client and STP are asked to provide the shared secret key as either party may be a suspect.

5. SECURITY CONSIDERATIONS

We have relaxed trust on the STP to not reveal customer PII and properly execute PII within the vendor business process activity. This opens up hostile STP possibilities, such as:

- STP falsifying the transaction outcome: client and vendor could request a dispute, resulting in the AA detecting an anomaly in the transaction;
- STP leaking vendor's business process activity: vendor can mitigate risk by code watermarking [Collberg and Thomborson, 2002] the business process activity for detection of misuse, such as disclosure or reverse-engineering;
- External denial of service (DoS) attacks: it is expected that the STP provides a list of replicated services to alleviate bottleneck and single point of failure concerns.

Collusion between two parties (for example, vendor and STP) prevents the remaining party from issuing a disputed transaction request. The remaining party could still contact an external certified entity (ECE) for further investigation.

We have assumed the STP will not knowingly disclose customer PII, however, in the case of compromise, a noticeable amount of information may accumulate over time. Customers can mitigate potential risk by choosing an STP that operates within the same data privacy laws and we expect that finding a reputable STP is easier than finding a reputable vendor.

Although privacy principles of 'data minimisation' and 'purpose binding' are not technically enforced by TEPS, compliance has been placed in the customer's domain. Customers can check vendor PII requests against their stated privacy policy before opting to continue with the transaction. Customers and STPs can check vendor purpose binding and is considered a legal issue if not followed, pre-empting a transaction dispute.

6. FORMAL ANALYSIS OF THE PROTOCOL

TEPS has been formally verified with the Casper protocol compiler and FDR2 model checker [Donovan et al., 1999] to prove confidentiality on customer PII data, vendor business process activities hold against all currently known communication channel attacks.

Due to combinatorial explosion of the search space, privacy assertions for composite web services could not be formally verified by FDR2. However, as simple web services privacy is formally verified, and composite web services are iterated simple web services, induction suggests TEPS provides technically-assured privacy of composite web services.

7. IMPLEMENTATION AND RESULTS

TEPS was implemented in Java with Web Services support for SOAP messaging and WSDL documents. Our system offers flexibility of public key certificate representations, supporting X.509 and SPKI/SDSI formats. X.509 is the industry standard, providing identity certificates but it requires hierarchies of fully-trusted certificate authorities and cannot handle threshold certificates. SPKI/SDSI is a simplified and flexible certificate system allowing identity and authorisation certificates, fine-grained access control and, most importantly, supports threshold certificates. We implemented a secured SPKI/SDSI framework, that was reported in [Pearce et al., 2004a, Pearce et al., 2004b], which allows for naming, access control and thresholding.

TEPS services use thread-based concurrency to support multiple transactions simultaneously. Business process activities (BPAs) are compiled Java bytecodes packaged as '.jar' archives. Vendors could possibly provide BPAs to semi-trusted processors in an encrypted form for confidentiality.

Experiments were conducted on Intel(R) PIII 1GHz machines, with separate machines for each service, communicating over a wired 100Mbps switched network. We measured client connectivity on both the 100Mbps switched network and a wireless 802.11g network at speeds of 1Mbps and 11Mbps. The wireless access point used media access control (MAC) filtering and Wired Equivalency Privacy (WEP) based encryption for additional security.

Table 1. Total client-wait times using TEPS with and without TLS

CONFIGURATION	TIME (sec)
TEPS, Wired 100Mbps	7.67
TEPS, Wired 100Mbps, SSL/TLS	8.35
TEPS, Wireless 1Mbps	8.01
TEPS, Wireless 11Mbps	7.67

Table 1 shows protocol performance in the client perspective by measuring total client-wait time over the entire length of a transaction. Vendor privacy policies and business process activities were fixed at one kilobyte each. Timing of business process execution by STPs were not performed as they gave a constant time among each experiment and, pragmatically speaking, are highly dependent on the business purpose of the vendor. Results from Table 1 indicate that TEPS is efficient at servicing simple web services transactions for both wired and wireless clients, with overheads of around seven to eight seconds per web services transaction. In fact, transaction times did not significantly differ for either wireless or wired network speeds, never exceeding 5% of total transaction times. This suggests that transaction performance will remain satisfactory as network speeds scale down further. Tunnelling TEPS over SSL/TLS incurred a penalty of nearly one second for total client wait-times. Service start-up times took an

Table 2. Processing and communications costs for participant

Party	Number of Messages		Total Message Sizes (kb)	Processing + Communication Times (sec)			Cryptographic Operations	
	Send	Recv		Send	Recv	Total	Encrypts	Decrypts
CL	4	4	~92	3.01	3.64	6.65	1 (symm)	1 (asymm)
V	3	4	~56	0.01	6.58	6.59	-	-
STP	5	3	~136	0.05	3.53	3.58	-	1 (symm)
CA	1	1	~60	3.09	0.04	3.13	3 (asymm sig)	-
AA	0	1	~0.8	-	0.11	0.11	-	-

additional three to five seconds for SSL/TLS enabled sockets due to key randomisation and secure socket establishment.

For a deeper understanding of practicalities within TEPS-enabled web services transactions, we measured processing and communication costs incurred by each party for each communicated message. This was collated to give an overview on how much work is performed by each participant, as shown in Table 2.

Client and vendor have the highest costs in terms of time, due to encryption, communication and awaiting responses from other parties respectively. The STP, as is evident with the vendor, spends almost all of its time waiting to receive messages, whereas the certificate authority incurs most of its costs in generating and communicating threshold certificates.

Our results suggest a linear extension of composite web services yields linear growth in time complexity. For example, the Health Clinic service detailed in Scenario 2 of Section 3 would involve three iterations of TEPS, each iteration being interleaved within its adjacent iteration with a total client wait-time approximately three times longer than a single iteration.

8. DISCUSSION AND FURTHER WORK

Through the use of a semi-trusted processor, TEPS guarantees protection of customer personally identifiable information (PII) against untrusted vendors in the application layer. This also prevents vendors from linking up databases and identifying customers on seemingly unlinkable attributes (triangulation). Introducing an accountability authority allows for externally certified entities to follow up unlawful activities.

TEPS supports execution of business process activities for (1) once-off transactions (for example, customer using an online broker) and (2) transactions requiring multi-vendor integration, that being composite web services.

In the first scenario, described in Section 3, the business process activity may require access to the vendor database (for example, an inventory table). It is the responsibility of vendor and semi-trusted processor to agree on appropriate mobile code and dependent parameters to satisfy business logic for execution of business process activities. One solution can involve the vendor attaching required data from its own database to the business process. Alternatively, both vendor and STP can agree on a common link for respective scrambled PII and plain-text PII database entries. The second scenario is addressed by iterating the TEPS protocol for each additional back-end vendor web service, creating a trusted chain for semi-trusted processors and an untrusted vendor chain. Complexity is linear which suggests that the system is extensible for transactions of growing numbers of interacting services. However, for large business processes or a large number of co-operating vendors, long running transactions (LRTs) may be required to provide acceptable client wait-time.

We expect to alleviate vendor reluctance of outsourcing full business processes to STPs by the use of code watermarking: detecting STP misuse, such as disclosure or reverse-engineering. More comprehensive solutions may be more applicable, such as source code escrow agreements.

TEPS prevents vendors from profiling clients, which is another privacy issue. However, if customers choose to allow profiling of their activities, the STP can profile customers based on gathered information, anonymise (by removing identifiable elements) and pass it back to the vendor.

We have not investigated programming challenges of aggregation and separation of business processes into activities that can be processed by separate parties. Furthermore, aggregation and separation of privacy policies among co-operating vendors is an area of future work.

Investigation into the benefits and trade-offs of caching vendor business policies with identity and authentication details will help decide whether additional performance gains are worth the risk against obsolescence. Vendor policies negotiated on a client-by-client basis presents an open problem in this approach.

9. CONCLUSION

In this paper we proposed the Technically-Enforceable Privacy and Security (TEPS) system that prevents vendors from ever obtaining customer personally identifiable information. Major components of the system were the following:

- semi-trusted processor to (1) protect customer personally identifiable information (PII) and (2) execute vendor-provided business processes with customer PII data in a protected environment;
- accountability service to provide recourse when one or more parties abuse the protocol;
- resolution mechanism for transaction disputes;

Furthermore, we showed how TEPS is extensible in supporting composite web services by iterating the protocol for multiple back-end vendors.

TEPS has been verified to ensure customer privacy is maintained against untrusted vendors or external attackers and that vendor business process activities are not accessible to parties other than the semi-trusted processor.

Our results indicated that the solution was suitable for web services as client wait-times for transactions were within an acceptable range. TEPS also performed well in slower wireless networks and transaction times grew in a linear fashion as complexity of interactions rose in composite web services scenarios.

TEPS gives privacy and security guarantees to prevent untrusted vendors from obtaining private customer information within traditional transactions and composite multi-vendor web services. In helping alleviate consumer concerns and address open issues of privacy within composite web services, service-oriented transactions can become a safer practice.

ACKNOWLEDGEMENTS

We would like to thank Formal Systems for providing a license to freely use FDR2. We would also like to thank the anonymous reviewers for their useful suggestions.

REFERENCES

[Ashley et al., 2003] Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. (2003). Enterprise Privacy Authorization Language (EPAL). Research Report 3485, IBM Research.

[Berthold and Köhntopp, 2001] Berthold, O. and Köhntopp, M. (2001). Identity management based on P3P. In *Lecture Notes in Computer Science*, volume 2009, pages 141–160.

[Campbell et al., 2002] Campbell, R., Al-Muhtadi, J., Naldurg, P., Sampemane, G., and Mickunas, M. Dennis (2002). Towards security and privacy for pervasive computing. In *Proceedings of the International Symposium on Software Security, Keio University*, Keio University, Tokyo, Japan.

[Casassa et al, 2003] M. Casassa Mont, S. Pearson, P. Bramhall. Towards Accountable Management of Privacy and Identity Information. ESORICS 2003: 146-161

[Chan et al., 2002] Chan, H., Lee, R., Dillon, T., and Chang, E. (2002). E-Commerce: Fundamentals and Applications. pages 287–298. ISBN: 0-471-49303-1.

[Chaum, 1981] Chaum, D. (1981). Untraceable Electronic Mail, Return Addresses and Digital Pseudonyums. *Communications of the ACM*, 24(2):84-90.

[Chaum, 1982] Chaum, D. (1982). Blind Signatures for Untraceable Payments. *Crypto*, pages 199–203.

[Chaum et al., 1990] Chaum, D., Fiat, A., and Naor, M. (1990). Untraceable electronic cash. *Proceedings on Advances in cryptology. California, United States*, pages 319–327.

[Collberg and Thomborson, 2002] Collberg, C. and Thomborson, C. (2002). Watermarking, Tamper-Proofing, and Obfuscation - Tools for Software Protection. In *IEEE Transactions on Software Engineering*, volume 28, pages 735–746.

[Dierks and Rescorla, 2004] Dierks, T. and Rescorla, E. (2004). The TLS Protocol Version 1.1. Internet Draft `http://www.potaroo.net/ietf/ids-wg-tls.html`.

[Dingledine et al., 2004] Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The Second-Generation Onion Router. In *In Proceedings of the 13th USENIX Security Symposium.*

[Donovan et al., 1999]Donovan, B., Norris, P., and Lowe, G. (1999). Analyzing a library of security protocols using Casper and FDR. In Workshop on Formal Methods and Security Protocols.

[Goldberg, 2002] Ian Goldberg. Privacy-enhancing Technologies for the Internet, II: Five Years Later. Workshop on Privacy Enhancing Technologies. April 2002

[internetnews.com, 2003] internetnews.com, Staff: (2003). Acxiom Hacked, Customer Information Exposed. Website: `www.internetnews.com/storage/article.php/2246461.`

[Jakobsson and Juels, 2001] Jakobsson, M. and Juels, A. (2001). An Optimally Robust Hybrid Mix Network. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing*, pages 284–292. ACM Press.

[Jendricke et al., 2004] Jendricke, U., Kreutzer, M., and Zugenmaier, A. (2004). Pervasive Privacy with Identity Management. In *Proceedings of ACM Symposium on Applied Computing*, pages 1593–1599. ACM Press.

[EU FP6 PRIME Project, 2005] PRIME: Privacy and Identity Management for Europe. Website: http://www.prime-project.eu.org/ Last accessed: 15-11-2004.

[Kenny and Korba, 2002] Kenny, S. and Korba, L. (2002). Applying digital rights management systems to privacy rights management. *Computers & Security*, 21(7):648–664.

[Medjahed et al., 2003] Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A. H. H., and Elmagarmid, A. K. (2003). Business-to-business interactions: issues and enabling technologies. *The International Journal on Very Large Data Bases*, 12(1):59–85.

[Pearce et al., 2004a] Pearce, C., Bertok, P., and Thevathayan, C. (2004a). A Protocol for Secrecy and Authentication within Proxy-Based SPKI/SDSI Mobile Networks. AusCERT Asia Pacific Information Technology Security Conference ISBN: 1864997745.

[Pearce et al., 2004b] Pearce, C., Ma, Y., and Bertok, P. (2004b). A Secure Communication Protocol for Ad-Hoc Wireless Sensor Networks. IEEE International Conference on Intelligent Sensors, Sensor Networks & Information Processions, Melbourne, Australia.

[W3C, 2002]W3C (2002). Platform for Privacy Preferences (P3P). W3C Recommendation `www.w3c.org/TR/2002/REC-P3P-20020416/.`

[Waldman et al., 2000] Waldman, M., Rubin, A., and Cranor, L. (2000). Publius: A robust, tamper-evident, censorship-resistant, web publishing system . In *Proc. 9th USENIX Security Symposium*, pages 59–72.