# POSTER: Re-Thinking Risks and Rewards for Trusted Third Parties

Jan-Ole Malchow, Benjamin Güldenring, Volker Roth
Freie Universität Berlin
Berlin, Germany
jan-ole.malchow@fu-berlin.de, benjamin.gueldenring@fu-berlin.de,
volker.roth@fu-berlin.de

## ABSTRACT

Commercial trusted third parties (TTPs) may increase their bottom line by watering down their validation procedures because they assume no liability for lapses of judgement. Consumers bear the risk of misplaced trust. Reputation loss is a weak deterrent for TTPs because consumers do not choose them – web shops and browser vendors do. At the same time, consumers are the source of income of these parties. Hence, risks and rewards are not well-aligned. Towards a better alignment, we explore the brokering of *connection insurances* and *transaction insurances,* where consumers get to choose their insurer. We lay out the principal idea how such a brokerage might work at a technical level with minimal interference with existing protocols and mechanisms, we analyze the security requirements and we propose techniques to meet these requirements.

## 1. INTRODUCTION

When buying goods and services on the Internet, consumers need to decide *who* they are buying from and *why* it is safe to buy from the remote party, particularly if it is an unfamiliar party. The first question is meant to be solved by means of transport layer security in conjunction with public key certificates issued by a mutually trusted *certification authority* (CA). *Trust seals* (TS) are meant to address the second question. However, CAs and TS providers are exposed to perverse incentive structures – they can increase their business and reduce their costs by watering down the standards of verification. This seems to happen in practice already [1, 6]. A recent study found that website with trust seals are not more secure than websites without [5]. The risks of CAs and TS providers are largely limited to reputation effects because they assume no liability for lapses of judgement. Reputation effects have limited power in this case because consumers do not participate in the decision which CA or TS provider a shop selects. They largely rely on the choices that have been made for them by the other parties [4]. Hence, consumers bear all the risk while paying directly or indirectly for the other parties. They may not pay money explicitly but their information and attention is marketed and this yields the income stream from which the other parties derive their compensation. In a functioning market, risks and rewards are aligned. Towards an improved alignment, we explore the *brokering* of *connection insurances* (CI) and *transaction insurances* (TI) as a means to solve the *who* and *why* questions, while giving consumers control over the choice of trust providers. We lay out the principal idea how such a brokerage might work at a technical level without interfering with existing protocols and mechanisms, we analyze the security requirements and we propose techniques to meet these requirements.



*CCS'16 October 24-28, 2016, Vienna, Austria*



ACM ISBN 978-1-4503-4139-4/16/10.

DOI: http://dx.doi.org/10.1145/2976749.2989060

## 2. ASSUMPTIONS AND MODEL

We assume that consumers install a client-side software and receive regular updates of a *insurance policy file.* The software produces cryptographic proofs of connections and transactions so that consumers can enforce contractual rights. Consumers choose their insurance company implicitly or explicitly when using this software. This allows consumers to be responsible market participants. An *insurance broker* (IB) mediates between consumers, website operators and insurers (IN). In most cases website operators will pay the insurance fee on behalf of the consumer – similar to how they currently pay CAs and TS providers. For privacy reasons our model also allows consumers to pay their insurance companies directly. Our protocols are designed to protect Alice and the IB from misbehavior of each other. For example, they provide:

**Connection insurance** (CI): "If the IB promised Alice that certificate $C$ is trustworthy at time $t$ and Alice established a connection to a web server who presented $C$ at time $t$ then Alice can provide convincing evidence of this fact to a judge."

Since we require that Alice's evidence is convincing this also protects IB and thus IN from fraud. Additionally, our system provides *reselling protection* (RP) to the IB. Without RP, Alice may sell her policy file to a third party, say, Charlie. This does not extend Alice's coverage to Charlie but it provides Charlie with protection against certificate substitutions. This results in a *free-riding problem* for the insurance market. While we cannot prevent Alice from sharing information she receives from the IB we can assure that this information is not convincing to a third party. Shops *do not have to change their software* to participate in this part of our system. If they wish to offer transaction insurances then they have to add software to their shop systems, similar to adding support for a new payment system. The technical details follow.

# 3. DESIGN OVERVIEW

Participating consumers receive a machine-readable *insurance policy* from the IB. In our current implementation, a consumer runs two browsers and a proxy software. One browser connects through the proxy to websites that are covered by the policy (CI). The second browser instance is for all other connections. The proxy verifies that HTTPS connections are covered by the policy and blocks all other connections.

If the user proceeds with a covered connection and suffers from covered damages then the proxy provides the evidence that is necessary to claim the insurance benefit. If the insurer disavows the claim then the user can take the evidence to the judiciary and prove that the connection was covered. The proxy approach yields compatibility with existing browsers. Using a separate browser instance with the proxy avoids problems associated with trust indicators and user decisions because the proxy only allows covered SSL/TLS connections and blocks all others. This eliminates phishing attacks as long as consumers stick to the proxied browser for their online shopping.

### *Connection Insurance.*

Our CI protocol involves a user *Alice* who visits web sites and who is insured by an *insurer* IN. The insurer relies on an *insurance broker* IB to maintain a list of trustworthy certificates. At the same time, IN provides certificates to the IB, which IN receives from the shops it insures. When Alice visits a web server *Bob,* information provided by the IB informs her whether the certificate $\mathrm{Cert}_{\mathrm{Bob}}$ presented by Bob is trustworthy or not. We additionally assume a trusted legal institution *Judge J* who is responsible for settling disputes between Alice, IN and IB and who enforces legal contracts signed by Alice and IN.

### *Transaction Insurance.*

The TI protocol involves a user *Alice* who transacts with a vendor *Bob*, and an *insurance broker* (IB) who issues and validates insurance vouchers. In order to insure a transaction, *Alice* submits a voucher to IB who validates it and sends an insurance policy back. Alice may receive vouchers from Bob or IN. If she receives the voucher through Bob then IB learns with whom Alice transacted. If Alice receives the voucher through IN the IB does not learn about Bob. Which way is chosen depends on who pays for the voucher. In the first case, Bob pays for the voucher and Alice receives the benefit but Alice gives up some privacy. In the second case, Alice pays for the voucher herself and keeps her privacy. While the TI protocol is a relatively straightforward application of digital signatures the CI protocol is more involved. Therefore we focus on the CI protocol in the following.

## 3.1 Insurance System

Our scheme uses a public key signature scheme $\Pi$, a collision resistant hash function $h$ and a cryptographic hash function $H$ modeled as random oracle. We write $\sigma_A(m)$ to denote a signature of $A$ on $m$ where $A$ may be a name (for example, Alice) or a certificate. If we write $\sigma_A(m, m')$ then this implies that $m$ and $m'$ are padded to suitable lengths and concatenated afterwards. Figure 1 gives a compact representation of the simplified scheme.

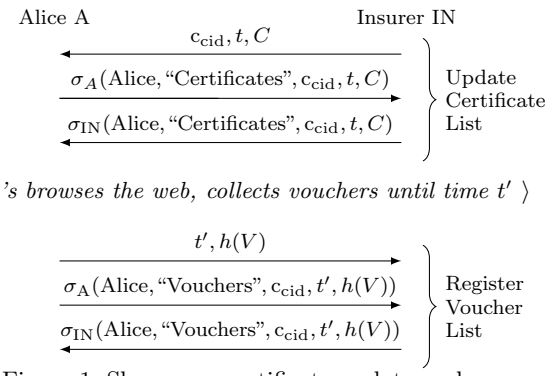


Figure 1: Shows one certificate update cycle.

### *Insurance Provider Setup.*

In order to set up service, the IB creates a signature key pair $(\mathrm{pk_{IN}}, \mathrm{sk_{IN}})$ and assembles an initial list of certificates $C := (\mathrm{Cert}_1, \dots)$ that it believes to be authentic.

### *User registration.*

In order to subscribe to the IB's service, Alice creates a signature key pair $(\mathrm{pk}_A, \mathrm{sk}_A)$ and her insurer generates a unique customer number for Alice. For simplicity we equate this number with "Alice". The insurance contract between Alice and the IB includes $(\mathrm{pk}_{IB}, \mathrm{pk_A})$, the customer number, the contract's validity term $(t_0, t_{\mathrm{end}})$ and an upper bound $\Delta T$ on the time between certificate updates.

### *Certificate List Download.*

The certificate list download marks the beginning of an update cycle, which is shown in Fig. 1. The IB sends a list of certificates $C$ and a randomly generated cycle identifier $\mathrm{c_{cid}}$ to Alice. Alice computes a signature on $\mathrm{c_{cid}}, C$, a timestamp $t$ and her customer number and sends the signature to the IB. The IB verifies Alice's signature, checks that the timestamp is recent and replies with a signature of the same message.

### *Voucher Submission.*

The voucher submission marks the end of an update cycle. Alice assembles a list of vouchers $V$ and sends the hash value $h(V)$ and a timestamp $t'$ to the IB together with a signature of these values, the current cycle identifier and her customer number. The IB checks the signature, that the timestamp is recent and that Alice did not violate the update interval requirement. The IB then replies with a signature of the same message.

### *Creating Vouchers.*

A voucher provides evidence of the fact that Alice established a connection to Bob's server at `bob.example.org` whose certificate $\mathrm{Cert_{Bob}}$ was included in certificate list $C$ in update cycle $\mathrm{c_{cid}}$. Alice computes vouchers as follows: she generates a fresh random value $r$, sets

$$v := \langle \text{Alice}, \texttt{bob.example.org}, \mathrm{c_{cid}}, r \rangle$$

and calculates $H(v)$. She sends $H(v)$ to Bob who replies with a signature $\sigma_{\mathrm{Bob}}(H(v))$. Alice stores the tuple

$$\mathrm{Vch} := \langle \mathrm{Cert_{Bob}}, v, \sigma_{\mathrm{Bob}}(H(v)) \rangle$$

where Vch is her voucher.

### *Demonstrating Insurance Cases.*

In order to demonstrate an insurance case, Alice has to prove three things: 1. $\text{Cert}_{\text{Bob}}$ was in the certificate list $C$ in cycle $\text{c}_{\text{cid}}$. 2. Alice updated her certificate list in time. 3. Alice connected to Bob's server in cycle $\text{c}_{\text{cid}}$. Alice proves the first two items by disclosing certificate list $C$, cycle id $\text{c}_{\text{cid}}$ timestamps $t, t'$ and the signatures

$$\sigma_{\text{IN}}(\text{Alice}, \text{“Certificates”}, \text{c}_{\text{cid}}, t, C),$$
$$\sigma_{\text{IN}}(\text{Alice}, \text{“Vouchers”}, \text{c}_{\text{cid}}, t', h(V))$$

issued by IB in cycle $\text{c}_{\text{cid}}$. Verifying that $\text{Cert}_{\text{Bob}} \in C$ is straightforward. Updates were timely if $t' - t \leq \Delta T$. The evidence is convincing because IB chose a unique cycle id $\text{c}_{\text{cid}}$ and Alice cannot compute the signatures herself without breaking the signature scheme. Neither can Alice reuse signatures issued to another customer because signatures of the IB include unique customer numbers. In order to prove the third item, Alice discloses the voucher list $V$ that matches hash value $h(V)$ and the corresponding voucher Vch. Verifying that $\text{Vch} \in V$ is straightforward. The proof is convincing because Alice would have to forge a valid signature of Bob or find a collision in one of the hash functions $h$ or $H$ in order to compute the evidence herself. Since $v$ includes her customer number she cannot reuse a voucher from another customer.

### *Adding Privacy.*

According to our preceding description of the protocol, Alice discloses her entire list of vouchers in order to claim compensation. This leaks connection information to the IB. We can solve this problem by making $h(V)$ the root of a Merkle tree. In order to prove that a voucher is a leaf $\ell$ of the tree, Alice discloses the path from the root to $\ell$. This still leaks the order of $\ell$ in $V$ and the size of $V$. We can avoid this leakage by Alice simply padding $V$ to the size of $C$ using pseudorandomly generated vouchers and randomizes the order in $V$.

### *Adding Reselling Protection.*

The basic scheme allows Alice to resell her certificate list to others. All she needs to do is disclose $C, t, \text{c}_{\text{cid}}$, her customer number and the signature of the IB. If Alice does not want to disclose the signature than she can still prove in zero-knowledge that she knows a valid signature. In order to remove this property, we require that Alice and the IB use a *Chameleon signature scheme* [3] in lieu of a regular signature scheme. Chameleon signature schemes have the property that Alice can convince herself that the IB signed a message but she cannot convince others that the IB is the signer because Alice could have forged the signature herself. The construction in [3] makes use of a Chameleon hash function $\mathcal{H}(\cdot, \cdot)$ in combination with a regular signature scheme.

If Alice presented such a forgery to a judge then the IB can produce a different message that yields the same hash value. Since $\mathcal{H}_{\text{Alice}}$ is considered collision-resistant without knowledge of the trapdoor information this indicates that Alice must have forged the signature. We incorporate the Chameleon signature scheme by replacing the third message in the certificate list download protocol with:

$$R, \sigma_{IB}(\text{Alice}, \text{“Certificates”}, \text{c}_{\text{cid}}, t, \mathcal{H}_{\text{Alice}}(R, C))$$

In order to uncover forgeries, the IB is now required to record all messages signed in this fashion. If the IB uses an authenticated channel to send this message or the certificate list to Alice then the authentication must be deniable.

### *Integration with TLS 1.2.*

We focus on the ephemeral key exchange methods `DHE_DSS` and `DHE_RSA` in RFC 5246 [2] because `DH_anon` does not authenticate the server and the non-ephemeral key exchange methods do not provide forward secrecy. The `ClientHello` message contains 28 bytes that the client chooses. This field is signed by the server and sent back to the client later on in the protocol. In the original voucher creation we store a signature $\sigma_{Bob}(H(v))$. If we sent the server a value $H(v)$ we get exactly this signature. We do so by choosing a suitable instantiation of a random oracle for $H$. The input to the random oracle is equivalent to $v$ used in the original voucher creation. Where $r$ is chosen uniformly at random and at least 28 bytes long.

## 4. CONCLUSION AND FUTURE WORK

We presented an insurance model for Internet connections and electronic commerce transactions and a system design that supports that model. Our design protects insurers and the insured from fraud by the other party. It offers reasonable privacy and specific features that cater to the peculiarities of electronic goods and services, in this case, reselling protection for value-added information. Furthermore, consumers are free to choose insurers they trust and who have a legal representation in a jurisdiction consumers trust. It is still a long way to a point where consumers might pay for their insurance directly but our proposal is a compromise. The costs of insurance are factored into the costs of products and services as is the case already for business related expenses such as contemporary trust seals, public key certificates and transport insurances. From our perspective, the most important property of our model is that it aligns revenue with risk – a property that is dearly missing in the current CA and trust seal ecosystem.

## 5. REFERENCES


[1] A. Arnbak, H. Asghari, M. Van Eeten, and N. Van Eijk. Security Collapse in the HTTPS Market. *Queue*, 12(8):30:30–30:43, Aug. 2014.

[2] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246, Aug. 2008.

[3] H. Krawczyk and T. Rabin. Chameleon Signatures. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2000, San Diego, California, USA*, 2000.

[4] S. B. Roosa and S. Schultze. The "Certificate Authority" trust model for SSL: a defective foundation for encrypted Web traffic and a legal quagmire. *Intellectual Property & Technology Law Journal*, 22(11):3, 2010.

[5] T. Van Goethem, F. Piessens, W. Joosen, and N. Nikiforakis. Clubbing seals: Exploring the ecosystem of third-party security seals. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, 2014.

[6] N. Vratonjic, J. Freudiger, V. Bindschaedler, and J.-P. Hubaux. The inconvenient truth about web certificates. In *Economics of information security and privacy iii*, pages 79–117. Springer, 2013.