A Protocol for Anonymously Establishing Digital Provenance in Reseller Chains

Ben Palmer, Kris Bubendorfer, and Ian Welch

School of Engineering and Computer Science
Victoria University of Wellington
Email: {ben,kris,ian}@ecs.vuw.ac.nz

Abstract. An increasing number of Internet traders exclusively sell digital products. These digital products can include media files, licenses, services, or subscriptions. We consider the concept of digital provenance in reseller chains. The goal of this work is to provide an honest customer with a guarantee on the origin and ownership history for a digital item *even when the reseller they are dealing with is untrusted*. We introduce a protocol called the Tagged Transaction protocol which uses a third party called the Tag Generation Centre (TGC) to provide a method for honest customers to check they are purchasing a legitimate item, anonymity for customers and resellers, a method for customers to resell items they have purchased to other customers, and verification of the TGC.

1 Introduction

Amazon, iTunes, and domain name resellers, such as GoDaddy, only exist as on-line traders with no physical stores. These digital products and services can include digital media or more abstract products such as an access 'right', a license, a service, or a subscription.

To check the origin of a digital item we need to provide the customer with a guarantee that the item has originally been purchased from the correct supplier. Checking the ownership history for a digital item involves checking that at every step in the reseller chain it has correctly been purchased and only sold on to one reseller or customer. We look at methods for establishing digital provenance anonymously to prevent any party involved in the protocol (or an observer) from building up detailed records of the identities of customers and resellers.

Most Internet resellers use a digital certificate to prove their identity and to provide information on their physical location and contact details. The digital certificate does not provide any mechanism or guarantees though which the provenance of goods might be established.

A simple approach to achieving digital provenance in reseller chains is to introduce a license server that acts as a trusted third party. This license server can check at every step in the transaction that the item is legitimate and that it has not been sold to multiple customers. This license server would then have control over a large amount of data both on the details of transactions conducted

and the identities of the parties involved. A better option is to provide verification of the actions of any third party in the protocol without reducing privacy.

To establish digital provenance anonymously in reseller chains we have developed a protocol called the Tagged Transaction Protocol. The protocol does *not* provide enforcement of licenses. The tagged transaction protocol uses 'tags' to establish provenance. If a reseller can provide a customer with a valid tag, the customer can have confidence that the reseller has sold them a properly licensed item. The tagged transaction protocol uses a Tag Generation Centre (TGC) to sign and check tags. The main four contributions of the Tagged Transaction protocol are:

1. a method for customers to check they are purchasing a legitimate item, even from an untrusted reseller
2. selectable anonymity for customers, resellers, and suppliers
3. mechanisms to verify the actions of the TGC so it is not required to be a trusted third party, and
4. customers can act in the role of a reseller and on-sell items.

2 The Tagged Transactions Protocol

The tagged transaction protocol provides a mechanism for establishment of the provenance of a digital item while preserving the anonymity of resellers and customers (and optionally suppliers). We use a Tag Generation Centre (TGC) to generate and sign tags. The tagged transaction protocol does not involve payment and we assume payment is made through an external third party.

2.1 Threat Model

A malicious reseller has several ways to try and defraud both the supplier and the customer. We have informally grouped these actions in to the following categories. Spoofing is where the reseller claims to be the supplier or tries to subvert the protocol to make it appear that they are the supplier. Counterfeiting is where the reseller sells the customer an item but never buys it from the supplier. Counterfeiting can be further divided into: fabrication where the reseller tries to forge a license for an item from scratch (or based on the structure of other licenses), cloning where the reseller tries to sell a license they have purchased from the supplier to multiple customers, and network sniffing where the reseller replays a legitimate license. We also class identity revelation where the customer learns the identity of one of the resellers (or optionally the supplier) that is not its neighbour in the chain as a category of attack.

We assume the reseller is a polynomially bounded active adversary. The customer and supplier are also assumed to be polynomially bounded. If the reseller is selling a customer item x, then we assume the reseller cannot collude with the supplier for x, but can try to impersonate the supplier for x. We assume the customer does not collude with the reseller.

2.2 Definitions and Techniques

The modified El-Gamal signature scheme for digital signatures is used because it has been proved secure in the random oracle model against adaptive chosen message attacks [1]. We use the notation of $\{A\}_{sk_B}$ to denote the message A signed using the key sk_B. All mathematical operations are computed modulo a large prime p in the group of integers $\mathbb{Z}_p$ closed under multiplication unless otherwise stated. We use the notation of pk to represent a public key and sk to represent a private key where $pk = g^{sk} \bmod p$. The value g is a generator for the group $\mathbb{Z}_p$ and q is some large prime where $q|p-1$ (q divides $p-1$). The parameters p, q, and g are global parameters. The TGC also generates its private key sk_{TGC} and public key $pk_{TGC} = g^{sk_{TGC}} \bmod p$.

A tag is a 4-tuple $\{A = pk_x, B = L_x, C = pk_{tag,r} = g^{sk_{tag,r}} \bmod p, D = a = g^z \bmod p\}$ with elements: $A = pk_x$ is the public key for the item, $B = L_x = \{id = H(x), tagno, License\}_{sk_x}$ is a license signed with the secret key for the item, $C = pk_{tag,r} = g^{sk_{tag,r}} \bmod p$ is the one time public key for the reseller r and tag tag, and $D = a = g^z \bmod p$ is the commitment value used in the zero knowledge proof of knowledge of the one time private key for the reseller.

The license $B = L_x = \{id = H(x), tagno, License\}_{sk_x}$ contains the identity of the item $id = H(x)$ and the unique tag number. The identity of the item $id = H(x)$ is calculated using a well known hash function H. To prevent the TGC being able to link actions done by a single reseller together, the reseller will use a separate one time private and public key pair for every tag. We denote this one time tag key as public key $pk_{tag,r}$ and secret key $sk_{tag,r}$ for reseller r and tag tag.

2.3 Stage 1 - Supplier Generating Tag with TGC

Before an item tag may be generated a one time registration phase must be completed. The supplier calculates the identity of the item $id = H(x)$ and a public (pk_x) and secret (sk_x) key for the item and registers the item and public key with the TGC. The TGC may convince itself with out of band checks that the party registering the item is indeed the rights owner. Where suppliers wish to remain anonymous, registration messages are sent via an anonymous communication channel. The TGC cannot verify ownership due to the anonymous channel and uses a first-in first-registered default. Anonymous channels are shown as the dotted lines in Figures 1 and 2.

The generation of a new tag for an item by the supplier takes place in the six steps shown in Figure 1, specifically: (1) The reseller sends a purchase request to the supplier containing the identity of the item they wish to purchase $id = H(x)$, the one time public key for the tag $pk_{tag,r}$, and a commitment value $a_r = g^{z_r} \bmod p$; (2) the supplier then creates a signed tag request containing the license L_x for the item signed using sk_x, the one time public key $pk_{tag,r}$, and the commitment value a_r all signed by sk_x; (3) the supplier sends the signed tag request to the TGC; (4) the TGC checks the tag number contained in the signed license $tagno$ has not been used for this item before and then constructs and

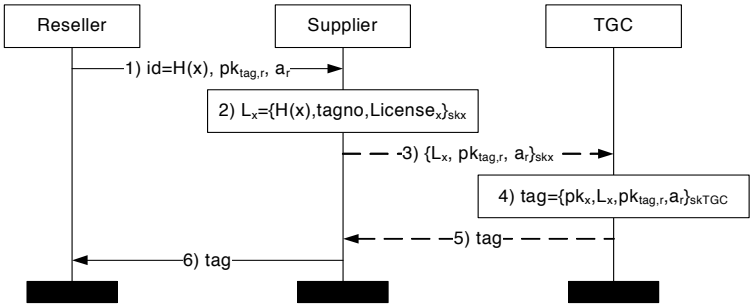

Fig. 1. Supplier Generating Tag with TGC

signs $tag = \{pk_x, L_x, pk_{tag,r}, a_r\}_{sk_{TGC}}$; (5) the tag is now sent from the TGC to the supplier; and (6) the supplier passes it on to the reseller. The reseller checks the tag has been signed by the TGC, that the license is for the correct item, and that the tag contains the correct one time public key and commitment value.

2.4 Stage 2 - Reseller Instantiating Tag with TGC

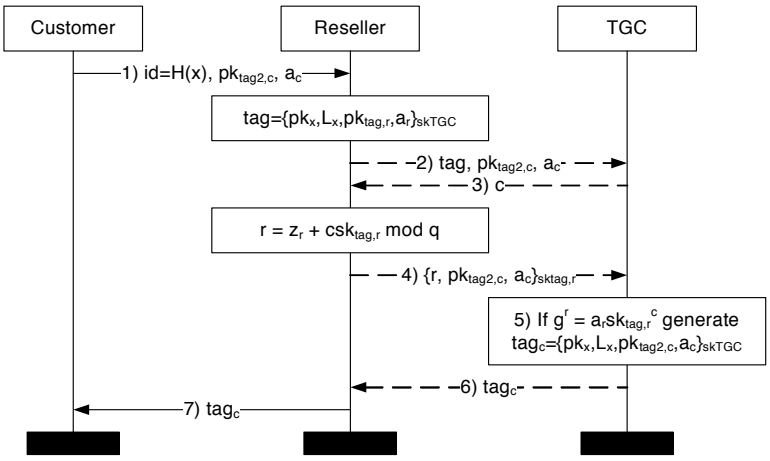

Fig. 2. Reseller Instantiating Tag with TGC.

The instantiation by a reseller of a new tag takes place in the seven steps shown in Figure 2, specifically: (1) the customer sends a purchase request comprised of the identity of the item they wish to purchase, the one time public key for the tag $pk_{tag,c}$, and a commitment value a_c; (2) the reseller sends the

one time public key for the tag $pk_{tag,c}$ and the commitment a_c to the TGC along with their tag for the item; (3) the TGC then checks that the one time public key has not previously been used for this item and tag number, the TGC and reseller then take part in a zero knowledge proof of knowledge of a discrete logarithm [2] where the prover (reseller) proves to the verifier (TGC) that they know the value $sk_{tag,r}$ such that $pk_{tag,r} = g^{sk_{tag,r}}$ using the commitment value $a_r = g^{z_r}$ (from the tag) and the TGC sends the challenge value c to the reseller; (4) the reseller calculates $r = z_r + csk_r \bmod q$ and sends $\{r, pk_{tag2,c}, a_c\}sk_{tag,r}$ to the TGC; (5) the TGC checks that the message is signed using $sk_{tag,r}$ and that $g^r = a_r pk_{tag,r}^c$ as $g^r = g^{z_r + csk_{tag,r} \bmod q} = g^{z_r} g^{csk_{tag,r}} = a_r pk_{tag,r}^c$.

The response value $r, pk_{tag2,c}, a_c$ is signed using the one time secret key for the tag $sk_{tag,r}$ to prove to the TGC that the reseller owns this tag. The TGC checks the tag sent to it by the reseller was signed using its private key sk_{TGC} and the values $pk_{tag2,c}, a_c$ and then uses the zero knowledge proof to detect if the tag has been replayed. If the tag has not been submitted to the TGC by a reseller before, the TGC saves the identity of the item $H(x)$, the public key $pk_{tag,r}$, the commitment a_r and the proof transcript c, r of the zero knowledge proof. If this tag is used a second time, the TGC will have two proof transcripts c_1, r_1 and c_2, r_2 that have been used with the same commitment value a_r. The TGC can then extract the one time secret key $sk_{tag,r}$ by computing $g^{r_1}/g^{r_2} = CB^{c_1}/CB^{c_2} = B^{c_1 - c_2}$ and $log_g B = r_1 - r_2/c_1 - c_2 = sk_{tag,r}$ The TGC can prove that the reseller has replayed the tag to any third party by presenting the two proof transcripts.

If the TGC does not detect replay, it generates a new tag that contains the public key $pk_{tag,c}$ and commitment value a_c of the customer or second reseller. This tag is then sent to the reseller (6) and then the reseller forwards it to the customer (7). The customer then checks that the tag was signed by the TGC and that the license L_x is valid and signed by sk_x. These seven steps can be repeated by the customer to resell this item to another party.

3 Security Analysis

Spoofing: An adversary could spoof the supplier in two ways. Firstly, it could forge a request from the supplier to create a new license. Forging a request for the supplier to the TGC is equivalent to the fabrication attack described below. Secondly, an adversary in control of the network could modify a valid registration message, to prevent this messages are encrypted using the public key of the TGC pk_{TGC}. The checks done by the TGC when a new item is registered should prevent an adversary from being able to register an item they do not control.

Fabrication: If the TGC receives a message signed by sk_x requesting a new tag that was not sent by the supplier, then either the adversary knows sk_x, or the adversary has been able to forge a message signed by sk_x, or the adversary is replaying a previously sent message. If the adversary replays a previously sent message, the TGC will not generate a new tag as the tag number for the item

has already been used. The chance of an adversary being able to forge a message signed by sk_x is negligible.

Cloning: If an adversary has run the protocol twice with the same input tag and produced two different tags tag_1 and tag_2 with different one time keys pk_1 and pk_2 and different commitments a_1 and a_2 where tag, tag_1, and tag_2 are all signed by the TGC then either the TGC has generated two tags or the reseller has tampered with tag to change the one time key or commitment value. The TGC will not generate two tags as it will be able to detect replay with two separate runs of the zero knowledge proof with the same one time key pk_{tag} and commitment a_{tag} and different challenge values c_1 and c_2. The chance of the reseller being able to tamper with tag to change the one time key or commitment value is negligible.

Network Sniffing: The adversary could intercept a tag that has already been generated by the TGC and send it to a customer or try and generate a new tag from the tag they have intercepted. In the first case, the chance of the tag containing the correct one time key pk_c and commitment value a_c chosen by the customer is negligible. In the second case, the adversary would have to be able to generate the message $\{r, pk_{tag2,c}, a_c\}_{sk_{tag,r}}$ and send it to the TGC as part of the zero knowledge proof. If the adversary does not know the secret key $sk_{tag,r}$ then the chance of it being able to create the message is negligible.

4 Anonymity and Verification of the TGC

We cannot always assume that our trusted TGC is trustworthy because it maybe subverted by the owners or third-parties through bribery, seizure or being compromised by attackers. An untrustworthy TGC could be used to reveal the identify of parties using the TGC or allow violation of the security properties provided by our protocol.

Identity can be protected by allowing communication with the TGC via an anonymity service such as TOR [3]. This would prevent even the TGC from knowing the identity of the suppliers and resellers limiting the amount of information it could maliciously reveal. A downside would be that selective revelation of identity could not be easily provided.

Verification that the TGC faithfully implements the security protocol can be achieved by requiring the TGC to publish all its actions to a public bulletin board. When a customer gets sent a tag from the reseller, it verifies that the tag has been correctly generated by the TGC by checking the operations the TGC has done to generate the tag. It also follows the actions of the TGC on the tag that the reseller had before generating the tag for the customer and so on down the chain of resellers until it reaches the initial tag creation by the supplier. Privacy is maintained because the only information leaked to the customer is the number of resellers the tag has passed through from the supplier to it.

5 Performance

The Tagged Transaction protocol makes use of digital signatures and zero knowledge proofs. We examine the computational complexity of the Tagged Transaction protocol based on the number of modular operations in Table 1 where n is the complexity of modular exponentiation and m is the complexity of modular division. This table does not take in to account any extra computations required to verify the actions of the TGC.

The Tagged Transaction protocol relies on the use of the TGC to generate and sign tags. As the actions of the TGC can be verified, there is no requirement for the TGC to be operated by a trusted party and we envisage many TGCs operating. The supplier for the item can select a TGC when it first generates the tag for an item based on previous relationships, results of verification of the TGCs past actions, availability, or some other metric.

Operation	Customer	Reseller	Supplier	TGC	Total
Generating Tag		$5n$	$5n + 2m$	$7n + m$	$17n + 3m$
Reseller Generating Tag	$8n$	$7n + m$		$9n + m$	$24n + 2m$
Verifying Tag	$6n$				$6n$

Table 1. Complexity of Operations in the Tagged Transaction Protocol

6 Related Work

The Paradiso system lets customer purchase not only the songs and videos from content providers but also reseller rights [4]. To prevent malicious behaviour, a Trusted Computing Module (TCM) is used to store encryption keys and to perform private key operations in secure memory while the Tagged Transaction Protocol does not rely on any trusted hardware.

The IEEE working group P1817 has produced a document suggesting a standard which is similar to the Paradiso system where customers can resell items they have purchased as they can with physical products [5]. While the P1817 standard does provide options for customers to resell content it relies on a trusted player to store cryptographic keys.

Serban, Chen, Zhang, and Minsky introduce the concept of a decentralised electronic marketplace (DEM) where transactions are subject to a set of trading rules [6] implemented using a mechanism called Law Governed Interaction (LGI). In LGI, a law is formulated using an event-condition-action pattern. Apart from the agents taking part in transactions in the marketplace, there are also a set of trusted controllers that enforce the law of the marketplace. Although these controllers are distributed, there is no method to verify their actions.

The Idemix system [7] developed by Camenisch and Van Herreweghen is an implementation of an anonymous credential system [8]. Both the Tagged Transaction protocol and one-show anonymous credentials provide replay detection but there are several differences. In anonymous credentials, when using multiple verifiers, replay detection is performed after the fact whereas in the Tagged

Transaction protocol it is done live by the TGC. In the Tagged Transaction protocol a reseller could transfer a tag from themselves to another reseller but this is not possible using anonymous credentials. The tagged transaction protocol also provides optional supplier anonymity and allows the resellers to instantiate tags when the supplier is offline neither of which are provided by anonymous credentials.

7 Conclusions and Future Work

In this paper we have presented the Tagged Transaction Protocol for establishing digital provenance anonymously in reseller chains. The tagged transaction protocol provides a method for honest customers to check they are purchasing a legitimate item, provides selective anonymity for customers and resellers (and optionally suppliers), provides mechanisms to verify the actions of the TGC, and allows customers to resell items. Future work in the tagged transaction protocol involves completing a security analysis using the FDR model checker. A further improvement of the protocol would be to prevent the leakage of information regarding the distance from the supplier of the reseller when the TGC is verified. We also intend to look for applications of the tagged transaction protocol in other areas.

References

1. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In: EUROCRYPT '96: Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology, Springer-Verlag (1996) 387–398
2. Schnorr, C.P.: Efficient identification and signatures for smart cards. In: EUROCRYPT '89: Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology, New York, NY, USA, Springer-Verlag New York, Inc. (1990) 688–689
3. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second-generation onion router. In: Proceedings of the 13th USENIX Security Symposium. (2004)
4. Nair, S.K., Popescu, B.C., Gamage, C., Crispo, B., Tanenbaum, A.S.: Enabling drm-preserving digital content redistribution. In: CEC '05: Proceedings of the Seventh IEEE International Conference on E-Commerce Technology, Washington, DC, USA, (2005) 151–158
5. P1817, I.W.G.: Initial technical description of the p1817 standard. Technical report, IEEE (2010)
6. Serban, C., Chen, Y., Zhang, W., Minsky, N.: The concept of decentralized and secure electronic marketplace. Electronic Commerce Research **8**(1-2) (2008) 79–101
7. Camenisch, J., Van Herreweghen, E.: Design and implementation of the idemix anonymous credential system. In: CCS '02: Proceedings of the 9th ACM conference on Computer and communications security, New York, NY, USA, (2002) 21–30
8. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, London, UK, Springer-Verlag (2001) 93–118