LAILA EL AIMANI & YONA RAEKOW (2010). Reselling Digital Content. In ARES 2010, LISA O'CONNER, editor, IEEE Computer Society, 391–396. IEEE Computer Society, 10662 Los Vaqueros Circle Los Alamitos, California 90720-1314. ISSN 0302-9743 (Print) 1611-3349 (Online). URL https://dx.doi.org/10.1109/ARES.2010.18.

Reselling Digital Content

Laila El Aimani, Yona Raekow
{*elaimani,raekow*}@*bit.uni-bonn.de*
b-it, Dahlmannstr. 2, Universität Bonn, 53113 Bonn, Germany

***Abstract*—Digital content, protected by specific terms of use, is currently delivered to customers via a few selected content providers. Allowing arbitrary entities, not just trusted content providers, to resell legitimately purchased, protected digital content to another entity, adds additional challenges to a DRM environment. In this paper, we formally model the problem of reselling digital content, and we provide a secure construction based on one-time (proxy) signatures. Our construction allows an arbitrary seller to resell its digital content to any buyer. We ensure that the identity of the buyer is only known to the seller. The buyer can verify that the purchased content is genuine. After the transaction is completed only the legitimate current owner can use the digital content. Any illegal use can be identified by a trusted authority.**

***Keywords*-Digital rights management, One-time (proxy) signatures, Anonymous communications, Secret sharing**

I. INTRODUCTION

Typically, digital content is made available to customers together with a license through content providers, e.g. Apple iTunes. Most of the times, a customer has the right to use this digital content based on certain rules specified by the license. In general, the user is not allowed to transfer or resell the digital content, even in case he has no use for it anymore or does not want it anymore. The scenario we consider in this work is that of digital artwork that shall not be made available to an unlimited amount of users (i.e. we do not consider the mass market of tunes and clips available in e.g. iTunes), but rather a small selected group. This can be compared to paintings, that have their value also due to the fact that usually only one of a kind exists. One could think of digital art pieces, like photographs, compositions of music or digital imagery that shall only become available in some private collections, showrooms or museums or in case of movies to a limited number of movie theaters. Museums and the like purchase the artwork that they would like to show usually for some period of time and then resell or transfer it to another museum. While there are well established procedures for "analog" art, like auctions for paintings, this process is somewhat more difficult when dealing with digital assets.

Another application of our scheme is the following: Typically one purchases a software for executing certain tasks. It is not uncommon that the software is only useful for some period of time, e.g. a computer game loses its value to his owner once he completed all levels. Currently there exist no legitimate way to resell the software to a third party. A third party could be interested in purchasing such a "used" software for the following reasons: The company that sold the original software does not sell the software anymore, went bankrupt or simply does not deliver the software to certain countries.

Allowing reselling adds additional challenges to the enforcement of digital rights: After the content is transferred to a new owner, the previous owner should not have the possibility to use the content any longer. In fact, nobody should be able to use the content, except the current, legitimate owner. The new owner wants to be sure that the content he bought is genuine and indeed from the artist it was claimed to be from. In this work we present a model that addresses these requirements and present a solution to the reselling digital content problem. Additionally our system keeps the new owner of the content anonymous. Only the previous owner knows the identity of the new owner.

In the following chapters we consider the following players: an artist *Art* (e.g. a photographer), who sells some images to *Sally*. After a while *Sally* no longer needs the artwork of *Art*, and wants to sell the images to *Bob* who is interested in having a genuine piece of art from the artist *Art*. Common practice in art sales is that the buyer *Bob* stays anonymous. Transferring digital content is prone to abuse from both *Sally* and *Bob*. *Sally* might be tempted to make copies and keep one to herself or even worse sell the same piece of art several times. This is due to the fact that copying digital assets is easy. *Bob* might want to share his newly acquired piece of art with all his friends and is tempted to provide them with the necessary access information.

In this paper we introduce a framework that allows reselling digital content such that *Bob* is only known to *Sally*. Only *Bob* can "use" the artwork and if the artwork is shared a trusted instance, e.g. a court can identify the illegitimate user. Furthermore when *Bob* buys the digital artwork from *Sally* he can verify that this piece of art indeed was produced by *Art* and not by somebody else. The scheme we propose relies on one-time proxy signatures [1], [2]. To the best of our knowledge one-time proxy signatures are used for the first time in order to develop a scheme for reselling digital content. The problem of reselling digital content has not been studied extensively. Previous work, e.g. [3], [4] developed schemes for "content redistribution", i.e. a user can distribute digital content in such a way that the DRM policies are preserved. We propose a scheme that allows content reselling, while ensuring that the seller has no longer the right to use or distribute the content once the transaction is made. We also ensure that the buyer stays anonoymous.

The rest of the paper is organized as follows: In section II we introduce the tools that we are using in our construction, followed by a description of a system for reselling digital content together with security requirements in section III. In section IV we present a construction of a system for reselling digital content. In the last section V, we conclude by discussing open questions and future work.

II. Preliminaries

In this section we will discuss briefly the tools that we are using to construct our system. We use one-time proxy signatures, anonymous communication and consumer compliant devices.

A. One-time digital signatures

One-time digital signature schemes can be used to sign, at most, one message; otherwise, signatures can be forged. A new public key is required for each message that is signed. One-time digital signature schemes have the advantage that signature generation and verification are very efficient. This is due to the fact that they rely on one-way functions without trapdoors. The following scheme is due to Lamport [5] and illustrates the idea of one-time signatures:

> **Key Generation:** For a given one-way function f, select two random strings x_0 and x_1 as private keys. Publish $f(x_0)$ and $f(x_1)$ as a one-time public key.
> **Signing:** Sign the bit $b \in \{0,1\}$ by revealing x_b.
> **Verifiying:** On input x'_b, check whether $f(x'_b) = f(x_b)$.

This scheme can be used to authenticate the signer to the verifier and corresponds to signing a 1-bit message, Merkle proposed a scheme that allows to send messages longer than one bit [6].

A one-time signature needs to satisfy the following requirements:

- **Unforgeability:** It is infeasible for any party not possessing the private key, to forge a message/signature pair that passes the signature verification.
- **Verifiability:** For a valid signature, a verifier is convinced the legitimate signer has signed the message.

B. One-time proxy signatures

One-time *proxy* signatures are one-time signatures with an additional proxy functionality: Proxy signatures allow a designated person, called a proxy, to sign on the behalf of a primary signer. A proxy signature convinces a verifier that the primary signer has delegated the signing power to the proxy and that the proxy has signed the message. A one time proxy signature scheme is usually comprised of the following phases [1], [2]:

> **Key Generation:** The primary signer and the proxy signer generate their private, public key pairs respectively. These key pairs can be used in a normal signature scheme.
> **Proxy Delegation:** The primary signer and the proxy signer execute an interactive protocol to generate a proxy public, private key pair $(\mathsf{pk}_p, \mathsf{sk}_p)$. sk_p is known only to the proxy signer and pk_p is public. Part of this interaction is the warrant m_w. The warrant specifies the delegation period, the identities of the primary and the proxy signer and a description of the message m to be signed.
> **Sign:** The proxy signer produces a signature σ on a message m using the proxy private key sk_p.
> **Verify:** The verifier, on input σ, m and pk_p, checks the validity of the signature using the verification equation.

For a one-time proxy signature scheme the requirements on unforgeability and verifiability need to be slightly modified, furthermore a one-time proxy signature scheme needs to satisfy also traceability:

- **Unforgeability:** It is infeasible for any party, not being the primary signer or the proxy signer, to forge a message/signature pair that passes the signature verification. I.e. a valid message signature pair can only be generated by the primary signer or the proxy signer.
- **Verifiability:** For a valid signature, a verifier is convinced that the primary signer has agreed to sign the message.
- **Traceability**: In case of a dispute between the primary and proxy signers, there exists a tracing algorithm that reveals the identity of the actual signer. I.e., the algorithm guarantees that it should be infeasible for:
  - the primary signer to sign a message m and to claim later that it has been signed by the proxy signer.
  - the proxy signer to sign a message and to claim later that it has been signed by the primary signer.

C. Anonymous Communication

Anonymous communication can be achieved through Onion Routing [7], which provides anonymous connections that are resistant to eavesdropping and traffic analysis. This is achieved by repeatedly encrypting a message and then sending it through several network nodes, the so called onion routers. It operates by dynamically building anonymous connections within a network of real-time Chaum Mixes [8]. Common anonymizing networks like e.g. the Tor network [9] introduce some delay into the communication (using a 2 Mbit/s connection, we achieved in tests a 1 Mbit/s throughput). Our construction does not suffer from this, since the amount of communication that requires anonymous communication is quite small (we send only several KBytes) and needs to be performed only once.

D. Compliant Device

We consider that every user has access to a so called *compliant device*. This can either be a specially designed hardware or a software, that adheres to certain rules and allows controlled use of given content. The device will only *play* the content, if all rules specified are fulfilled. Compliant devices are common when using DRM protected digital content.

E. Preventing Double Playing

One major problem when moving away from a centrally controlled content distribution system to a system where every user is allowed to redistribute content is that of *double play*, meaning that a user redistributes its content, but also keeps a copy for himself. Or a user sells the same asset more than once. In order to identify users who are giving content they purchased together with their corresponding credentials to unauthorized users, we will make use of Shamir's secret sharing [10]. The following scheme has already been deployed in e-cash settings (e.g. [11]) in order to prevent double spending of electronic coins.

Setup: Every user in the system has a secret $\mathsf{sk} = (a, b)$.

Commit: When the user plays some content for the first time on a given compliant device, the device transmits some information about the asset to be played as well as (x_0, y_0), s.t. x_0 is a challenge and $y_0 = ax_0 + b$ to a trusted authority.

Trace: If the Trusted Authority receives for the same item (x_0, y_0) and (x_1, y_1) with $x_0 \neq x_1$ it reveals the users identity.

If a user is caught cheating then his public key is removed from the system and he cannot participate anymore, i.e. he cannot play back his content nor redistribute content.

III. Reselling Digital Content

In this section we first introduce the framework, i.e. formally define the parties involved in a Reselling Digital Content Scheme and how they interact with each other. Afterwards we define the requirements that a Reselling Digital Content Scheme has to fulfill.

A. Framework

The reselling digital content problem involves a set of playing parties; the first entity in this framework is the digital content intended to be sold or resold, then there is the artist, who is the author of the content, and the buyer who purchases the content either directly from the artist or from another buyer. Finally, and in order to ensure the sound behavior of these parties, further entities are needed such as a trusted authority (TA) and a compliant device designated to play a genuinely purchased product. An overview of the interaction of these playing parties is depicted in Figure III-A.

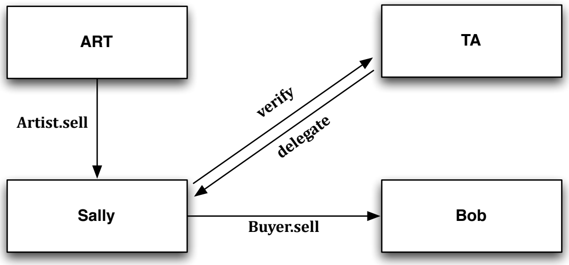

Figure 1. Entities in the reselling digital content problem

- **TA:** Refers to the trusted authority who is meant to trace back a dishonest user. The TA enables also a genuine owner of a content to resell it.
- **Content:** It represents the digital content intended to be sold or resold. Content has a unique identifier known publicly. In real life, the content may be a picture, a movie, a song, a software, etc...
- **Device:** This is a software or hardware designated to "play" a genuinely purchased product. For this purpose, it is supplied with an "online" access to the TA in order to check the legitimacy of a given Content or its ownership. We stress however that during this Device-TA communication, the privacy of the current owner is preserved.
- **Artist:** This is the author of the content. The Artist must be able to sell his products to any user of the system.
- **Buyer:** It denotes the entity interested in purchasing the digital content. The Buyer can buy the product from the artist or from another buyer, who is currently owning the product.

A Reselling Digital Content Scheme (RDCS) consists of the following algorithms:

Setup: This is a multi-party protocol between all artists and the trusted authority TA. In this protocol, the artists register the identifiers of their products (digital contents). At the end of the protocol, every content is bound to a unique identifier, and this information is publicly availble.

Key Generation: Refers to the key generation algorithm. On the input security parameter κ, the algorithm outputs a key pair $(\mathsf{pk}_{\mathsf{TA}}, \mathsf{sk}_{\mathsf{TA}})$ consisting of the public key and the private key of the trusted authority TA. Moreover, the algorithm outputs also key pairs $(\mathsf{pk}_{u_i}, \mathsf{sk}_{u_i})$, where u_i is an artist or a potential buyer. A list of content identifiers along with the public key of the corresponding artist is publicly available.

Artist.sell: This is an algorithm run by Artist to sell Content to a buyer given by pk_b. The algorithm inputs Content, the public key pk_b and the private key sk_a of Artist (author of Content), and output a string, called *ownership*, that proves that the buyer, whose public key is pk_b, is the owner of Content.

Verify: This is an algorithm run by any user of the system that checks the validity of an ownership on a given Content. The algorithm inputs Content, a public key pk_b of the alleged owner and his ownership. The result, computed with respect to the public verification key pk_p, is a boolean 1 if the ownership is valid and 0 otherwise. If Content is sold for the first time then the verification is done w.r.t. the public key of Artist pk_a

Delegate: This is a protocol between the TA and the Buyer (given by pk_b), owner of Content. The common input of the protocol is Content and its ownership. At the end of the protocol, the Buyer gets a private (proxy) key sk_p enabling him to resell Content, and a public (proxy) key pk_p for verification.

Buyer.sell: This is an algorithm run by a Buyer (different from Artist), given by pk_b , of Content in order to change the ownership of Content to another Buyer. More precisely, the algorithm inputs Content, its current owner (given by pk_b) together with his ownership and a private key sk_p (obtained from TA after the delegate protocol), and finally the potential new Buyer given by $\mathsf{pk}_{b'}$. The output is a string representing the new ownership of Content by the new Buyer whose key is $\mathsf{pk}_{b'}$.

Authenticate: This is an algorithm run by a user in the system that proves his ownership of some digital content. The algorithm inputs Content and the public key pk_b of the Buyer (presumed owner) and returns a string, representing the ownership of Content by pk_b, which can be using [verify].

Play: This is an algorithm run by Device. The algorithm inputs Content, an ownership w.r.t. pk_b and the private key sk_b corresponding to pk_b of the current owner. The result, computed w.r.t. a public verification key pk_p, is a boolean 1 if Content is correctly purchased and 0 otherwise.

Trace: This algorithm is run by the TA to trace a dishonest user. The algorithm inputs a product identifier and outputs the private key of the misbehaving user.

B. Security Model

A reselling digital content scheme (RDCS) should satisfy the following completeness properties.

1) Artist must be able to sell his products.

$$\forall\ \mathsf{Content}\ \ \forall\ \mathsf{pk}_b\ \ \forall\ (\mathsf{pk}_a, \mathsf{sk}_a) : \text{if } o = \mathsf{Artist.sell}(\mathsf{Content}, \mathsf{pk}_b, \mathsf{sk}_a)$$
$$\text{then } \mathsf{verify}_{\mathsf{pk}_a}(\mathsf{Content}, \mathsf{pk}_b, o) = 1$$

This means if Content was sold using the Artist.sell algorithm properly, which produces the string o then the verification algorithm should correctly verify the ownership o.

2) A genuine Buyer, owner of a content, must be able to resell it.

$$\forall\ \mathsf{Content}\ \ \forall\ \mathsf{pk}_b : \text{if } \sigma = \mathsf{authenticate}(\mathsf{Content}, \mathsf{pk}_b) \text{ and}$$
$$(\mathsf{sk}_p, \mathsf{pk}_p) = \mathsf{delegate}(\mathsf{Content}, \mathsf{pk}_b, \sigma) \text{ then}$$
$$\mathsf{verify}_{\mathsf{pk}_{p_b}}(\mathsf{Content}, \mathsf{pk}_b, \sigma) = 1 \Rightarrow$$
$$\mathsf{verify}_{\mathsf{pk}_p}(\mathsf{Content}, \mathsf{pk}_{b'}, \mathsf{Buyer.sell}(\mathsf{Content}, \mathsf{pk}_b, \sigma, \mathsf{sk}_p, \mathsf{pk}_{b'})) = 1$$

where pk_{p_b} is the current verification public key of Content. Given that a buyer can prove that he is the legitimate owner and the delegation key pair was correctly computed using the delegate protocol then, in case the verification algorithm is successful, the verification shall be successful after the content has been transferred to a new owner, i.e. the new owner can be from now on verified as the legitimate owner.

3) A device must always play a correctly purchased item.

$$\forall\ \mathsf{Content}\ \ \forall\ (\mathsf{pk}_b, \mathsf{sk}_b)\ :$$
$$\text{if } \sigma = \mathsf{authenticate}(\mathsf{Content}, \mathsf{pk}_b) \text{ then}$$
$$\mathsf{verify}_{\mathsf{pk}_p}(\mathsf{Content}, \mathsf{pk}_b, \sigma) = 1 \Rightarrow \mathsf{play}_{\mathsf{pk}_p}(\mathsf{Content}, \mathsf{sk}_b, \sigma) = 1$$

If a user can authenticate himself and prove that he is the legitimate owner, then the verification algorithm will be successful and this leads to a successful rendering of the Content by device.

Moreover, a RDCS must also enjoy the following security properties.

1) **Unforgeability for the Artist.** Artist cannot sell a content of which he is not author. Also, Artist should not sell twice the same Content with the same ID.
2) **Unforgeability for the Buyer.** The Buyer cannot sell a content that he does not own. Moreover, the Buyer cannot sell more than once a content that he genuinely owns.
3) **Traceability.** A Buyer, who shares Content along with the necessary information for playing it with other users, must have his identity revealed by the TA.
4) **Privacy.** The identity of the current owner of a given Content is kept private.
5) **Deniability.** The TA cannot falsely accuse an honest Buyer of selling Content.

IV. A Construction Based on One-Time Proxy Signatures

In this section we present a solution to the reselling digital content problem which is secure in the model defined in Section III. Our solution involves many well known cryptographic mechanisms, described in Section II. In a nutshell, to sell Content, Artist will produce a one-time signature on Content concatenated with the public key pk_b of the Buyer. The resulting signature forms the ownership of the Buyer on Content. Later, if the Buyer wants to resell Content, he contacts the TA and gets the possibility of selling again Content, namely, he gets a key to produce a one-time proxy signature on the same product concatenated with the public key of the new buyer. To ensure the good working of all entities, a consumer compliant Device is in place; the Device plays Content owned by a Buyer, given by the public key pk_b, if the Buyer possesses a valid ownership of Content. Moreover the Device reveals half of the private key sk_b of Buyer anonymously to the TA, such that if the Buyer shares his ownership and private key with other users, the TA will be able to trace him. A detailed description of the construction is depicted below.

Setup: In the course of this protocol, the artists register their Contents, and get in response identifiers that uniquely bind the products. The function that binds a content to an identifier can be implemented for instance by a secure cryptographic hash function. This protocol also invokes the setup algorithms/protocols of the different components that will be used, namely the setup algorithm of a one-time signature scheme Σ and the setup algorithm of a one-time proxy signature scheme $\tilde{\Sigma}$ and finally the setup algorithm of an anonymous communication network.

Key Generation: In this algorithm, we invoke $\Sigma.\mathsf{keygen}$ for every Artist to generate a pair $(\mathsf{pk}_a, \mathsf{sk}_a)$ of public and private key. The algorithm $\tilde{\Sigma}.\mathsf{keygen}$ is called to generate a key pair $(\mathsf{pk}_{\mathsf{TA}}, \mathsf{sk}_{\mathsf{TA}})$ for the trusted authority TA and a key pair $(\mathsf{pk}_b, \mathsf{sk}_b)$ for every potential Buyer [1]. We assume that sk_b is given by a pair (s, t) where s represents the slope of a line and t its offset. Finally, a list $\mathcal{L}$ (maintained by the TA) of records is made public, where each record has as first entry the Content identifier, the second entry consists of the public key of the corresponding Artist and finally a third and fourth entry which are initially empty (they will later be updated with the public key of the proxy signature used to sell the content and a *commitment* of the current owner once he plays the content for the first time).

Artist.sell: To sell a Content c to a Buyer who is given by the public key pk_b, Artist produces a one-time signature $\sigma = \Sigma.\mathsf{sign}_{\mathsf{sk}_a}(c \| \mathsf{pk}_b)$ on $c \| \mathsf{pk}_b$. σ forms the *ownership* of Buyer with respect to c. A one-time signature ensures that Artist sells the digital content Content c only once. Having this one-time proxy signature produced on $(c \| \mathsf{pk}_b)$ indicates that the Buyer given by pk_b is the current owner of Content c.

Delegate: In this protocol, the TA checks the validity of the ownership of the alleged Buyer w.r.t. Content (see the algorithm verify). In case it is valid, the TA and the Buyer invoke the $\tilde{\Sigma}.\mathsf{delegate}$ protocol to generate a proxy key pair $(\mathsf{pk}_p, \mathsf{sk}_p)$, where sk_p is known only to Buyer and will serve him resell Content. Using one-time proxy signatures enables an owner to resell Content legitimately exactly once, without resorting to TA. Note that the warrant in this protocol is Content. Moreover, the record in the list $\mathcal{L}$ which corresponds to the identifier of Content will have the third entry updated with pk_p.

Buyer.sell: This algorithm is performed by pk_b (Buyer with public key pk_b) to resell a Content c. It produces a one-time proxy signature, using the proxy private key sk_p obtained after the execution of the delegate protocol, on $c \| \mathsf{pk}_{b'}$, where $\mathsf{pk}_{b'}$ is the public key of the new buyer. The resulting signature forms the ownership of the new buyer. Note that this new buyer can check whether the holder of c is a genuine owner by simply checking the validity of his ownership using the algorithm verify.

Verify: To verify a presumed ownership σ of a Buyer, given by pk_b, w.r.t. a Content c, a user will proceed as follows. Let $(i_c, \mathsf{pk}_a, \mathsf{pk}_p, -)$ be the record of the list $\mathcal{L}$ which corresponds to c. In case pk_p is not defined, σ is valid if $\Sigma.\mathsf{verify}_{\mathsf{pk}_a}(\sigma, c \| \mathsf{pk}_b) = 1$. If pk_p is defined, σ is valid if $\tilde{\Sigma}.\mathsf{verify}_{\mathsf{pk}_p}(\sigma, c \| pk_b) = 1$.

Authenticate: A Buyer, who claims to be the owner of Content, runs this algorithm to produce a string, representing his ownership, on which the algorithm verify outputs 1.

Play: Whenever the Device has to play a Content for a Buyer given by $(\mathsf{pk}_b, \mathsf{sk}_b)$, it checks the validity of the ownership of pk_b, using verify (we assume that the Device can access the list $\mathcal{L}$ anonymously). Furthermore, if the Device plays for the first time, it generates a random x_o (x-coordinate) and produces y_0 such that the point (x_0, y_0) belongs to the line sk_b. Finally, the Device sends *anonymously* (x_0, y_0), together with the identifier of Content, to the TA, who updates the fourth entry of the $\mathcal{L}$ record corresponding to Content with (x_0, y_0). Finally, the algorithm play returns 1 if all steps are executed successfully.

Trace: This algorithm inputs two points (x_0, y_0) and (x_1, y_1) and returns the line sk_b that connects those two points.

[1] Note that the one-time proxy keys $(\mathsf{pk}_p, \mathsf{sk}_p)$ are generated in the **Delegate** protocol

A. Analysis

We note that the completeness properties, defined in III-B are met as a direct consequence of the completeness properties of the underlying building blocks, i.e. the one-time (proxy) signature schemes and the anonymous communication network. Moreover, we claim that our construction is also secure.

1) **Unforgeability for the Artist.** This is ensured by the use of the one-time signature. In fact, Artist cannot sell an item that is not his, otherwise this would correspond to a forgery of the signature scheme. Moreover, he cannot sell more than once a given $\mathsf{Content}$ due to the one-time property of the signature scheme.
2) **Unforgeability for the Buyer.** This is ensured by the unforgeability of the used one-time proxy signature.
3) **Traceability.** If a genuine user shares his valid ownership and private key with other users in order to allow them to play a given content. Then, his private key sk_b will be revealed as soon as the content is played on another device. In fact, if the TA has to update the fourth component of a record which already contains a point, then the TA will be able to recover the private key sk_b of the misbehaving user since sk_b is nothing but a line which is reconstructable by two different points. We must highlight here that the random generation of the point's x-coordinate by Device (see the algorithm play) ensures having different points with high probability.
4) **Privacy.** It is guaranteed by the use of anonymous communications.
5) **Deniability.** This is guaranteed by the traceability property of the one-time proxy signature.

V. Conclusion

We presented a model for the reselling digital content problem, along a construction that preserves the anonymity of the buyer. To the best of our knowledge, this is the first proposal that transfers ownership of digital content, such that the new owner stays anonymous. In future work, we will investigate whether it is possible to provide a construction that does not rely on a trusted authority at all. When implementing this scheme in practice it is necessary to define clearly properties of the compliant device, such as access control, tamper resistance. A weakness in our scheme is that the owner of content needs to establish a connection to the trusted authority before rendering the content. We believe that this does not result in a major problem when considering the applications mentioned in the discussion, namely a museum displaying digital art, or the use of software licenses. We also note that currently more and more mobile devices (like mp3 players) are equipped with network access. Common practice in software licenses is also that a check is not performerd *every time* a license is used but in fixed time intervals, e.g. every day once. Since most mobile devices are synchronized with PCs that have network access, the online check can be performed when synchronizing on a daily basis. Our construction allows to play the content on one compliant device. This can be easily extended to more devices by having the secret key sk of a user represent the coefficients of a polynomial of degree > 2. We did not discuss how the buyer transfers money to the seller. We assume that this can be done via standard payment methods, e.g. PayPal.

VI. Acknowledgements

We thank the ARES 2010 reviewers for their valuable comments. This work was funded by the B-IT Foundation and the Land Nordrhein-Westfalen.

References

[1] Young-Seol Kim and Jik Hyun Chang, "New one time proxy signature scheme based on dlp using the warrant," *International Journal of Computer Science and Network Security*, vol. 7, no. 2, pp. 215–220, 2007.

[2] Huaxiong Wang and Josef Pieprzyk, "Efficient one-time proxy signatures," in *ASIACRYPT*, Chi-Sung Laih, Ed. 2003, vol. 2894 of *LNCS*, pp. 507–522, Springer.

[3] Srijith Krishnan Nair, Bogdan C. Popescu, Chandana Gamage, Bruno Crispo, and Andrew S. Tanenbaum, "Enabling drm-preserving digital content redistribution," in *CEC*, 2005, pp. 151–158.

[4] Claudia Eckert, Omid Tafreschi, and Frederic Stumpf, "On controlled sharing of virtual goods," in *In Proceedings of the 7th International Workshop for Technical, Economic and Legal Aspects of Business Models for Virtual Goods*, 2009.

[5] L. Lamport, "Constructing digital signatures from a one-way function," Tech. Rep., October 1979.

[6] Ralph C. Merkle, "A digital signature based on a conventional encryption function," in *CRYPTO*, Carl Pomerance, Ed. 1987, vol. 293 of *LNCS*, pp. 369–378, Springer.

[7] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson, "Onion routing," *Commun. ACM*, vol. 42, no. 2, pp. 39–41, 1999.

[8] David Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–88, 1981.

[9] Roger Dingledine, Nick Mathewson, and Paul F. Syverson, "Tor: The second-generation onion router," in *USENIX Security Symposium*. 2004, pp. 303–320, USENIX.

[10] Adi Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[11] Niels Ferguson, "Single term off-line coins," in *EUROCRYPT*, 1993, pp. 318–328.