An Identity Based Proxy Re-Signature Scheme

Tulasi Menon

***Abstract*—A proxy re-signature scheme is one which allows a semi-trusted proxy to transform a signer A's signature on a message m, into a signature of signer B on the same message. The proxy is however not allowed to sign arbitrary messages on behalf of either signer. Several proxy re-signature schemes have been proposed, including one ID-based scheme, by Hu et al. In this paper, we describe the flaw in this scheme, with respect to the definitions of delegator and delegate security defined by Ateniese et al. We define a novel and secure ID-Based proxy re-signature scheme, which satisfies these security definitions. In addition, we explore the applications of such a signature in inter-domain security. . A modified version of this signature, which can be used to transform signatures between signers in different ID-Based domains, is detailed.**

***Index Terms*—Identity based signature, inter-domain, Proxy re-signature.**

I. Introduction

In 1998, Blaze et al.[2] first introduced the primitive of Proxy re-signatures at Eurocrypt '98 and defined. In a proxy re-signature scheme, a semi-trusted proxy acts as a translator between two users, Alice and Bob, to transform a signature from Alice (the delegatee) into a signature from Bob(the delegator), on the same message. There followed a period where this primitive was largely ignored, until Ateniese et al [3] reopened the discussion in 2005, with new security definitions, a formal model and two new algorithms and proved their security in the random oracle model. More recently, in 2007, Shao et al [4] proposed the first scheme that claimed to be secure in the standard model. However, in 2008, S.Chow et al [6], performed a cryptanalysis of the scheme by Shao and proposed a new proxy re-signature scheme secure in the standard model, based on homomorphic signatures. The first ID-based version of this signature was proposed by Hu et al.[5] and is claimed to be secure in the standard model. Proxy re-signatures can be classified as unidirectional or bidirectional signatures. In a unidirectional proxy re-signature, the re-signature key allows the proxy to convert Alice's signature to Bob's, but not Bob's to Alice's. This property is useful in applications where the trust relationship between the two parties is not mutual. Schemes that do not satisfy this property are bidirectional signatures.

A Proxy re-signature must satisfy the following internal security claims, as discussed in [3]:

Limited Proxy: If the delegator and delegatee are both honest, then the proxy (1) cannot produce signatures for the delegator on any message other than the ones previously signed by the delegate and (2) cannot produce any signatures for the delegatee.

Delegatee Security: If the delegatee is honest, then he is safe from a colluding delegator and proxy. That is, even if the delegator and the proxy work together, they will not be able to forge the delegatee's signature.

Delegator Security: If the delegator is honest, then he is safe from a colluding delegatee and proxy. That is, even if the delegatee and the proxy work together, they will not be able to forge the delegator's signature on a new message.

A. Applications

There are several possible applications for proxy re-signatures, as have been listed in the existing literature. One possible application is to introduce anonymity. Individual signatures can be transformed to group signatures, by using the key of the group manager, or a common group key. The advantage here is that the proxy does not know either of the keys, and thus the group manager's key remains private even if the proxy is compromised. They also aid in the usage of machine readable travel documents like e-passports. At each point of travel, the signature within the passport can be transformed through different checkpoints, so that only one transformable signature needs to be held at a time. Another unique application of this signature is in inter-domain security. There are many cases, when a signature in one domain needs to be transformed into a signature by a party in another domain. This can happen in cases such as, mergers and takeovers of companies, and during cross-certification checks. In these cases, the delegator and delegate are not within the same domain, and the proxy is responsible for transforming the signature and rendering it readable in the delegatee's domain. Modifying the signature to achieve inter-domain operability has been illustrated later in the paper.

B. Our Contribution

In this paper, we perform a cryptanalysis of Hu et al's scheme and point out the flaws with respect to the definitions of delegator and delegate security. In order to correct this, we propose a new unidirectional signature that passes these definitions, and is computationally more efficient than that of Hu et al [5]. We provide analytical proofs of unforgeabilty with regard to the three different types of adversaries. A modified version of our signature is also described, which can be used to ensure inter-domain security.

II. Preliminaries

A. Bilinear Pairings

Let $<G_1, +>$ be a cyclic additive group generated by *P,* whose order is a large prime *q,* $<G_2, \cdot>$ be a cyclic

Manuscript received April 9, 2012; revised May 2, 2012.

Tulasi Menon is with the Tata Consultancy Services (TCS) Ltd,Chennai, India (e-mail: tulasi.menon@tcs.com).

multiplicative group of the same order, and let e: $G_1 \times G_1 \rightarrow G_2$ be a bilinear pairing with the following properties:

1. Bilinear: For any $Q,R,T \in G_1$, $e(Q+R,T)=e(Q,T)\cdot e(R,T)$ and $e(Q,R+T)=e(Q,R)\cdot e(Q,T)$

2. Non-degenerate: There exists R, $T \in G_1$, such that $e(R, T) \neq 1$

3. Computable: There exists an efficient algorithm to compute $e(R, T)$ for any R, $T \in G_1$.

B. Hard Problems:

Definition 1: Computational Diffie-Hellman (CDH) Problem in (G_1, G_2):

Given P, $a\cdot P$, $b\cdot P \in G_1$ for some unknown a, $b \in Zq$, compute $abP \in G_1$.

Definition 2: Decisional Bilinear Diffie-Hellman (DBDH) Problem in (G_1, G_2):

Given $P,a\cdot P,b\cdot P,c\cdot P \in G_1$ for some unknown a, b, $c \in Z_q$, compute $e(P,P)^{abc} \in G_2$.

III. Framework of an ID-Based Proxy Re-signature

A unidirectional-ID-based proxy re-signature scheme consists of six algorithms: Setup, Extract, ReKeyGen, Sign, ReSign and Verify.

Setup: Input a security parameter k, return the public parameters *params*, and keep the master secret *msk* to itself.

Extract: Input the public parameters *params*, an identity *ID* and the master secret *msk*, return the private key S_{ID} of *ID*.

ReKeyGen: Input the public parameters p*arams*, two identities ID_1, ID_2, and two private keys S_{ID1} and S_{ID2} corresponding to the identities ID_1, ID_2 respectively, return a re-signature key $rk_{ID1\rightarrow ID2}$.

Sign: Input the public parameters *params*, an identity ID_1, S_{ID1} and a message m, output a signature σ.

ReSign: Input (ID_1, ID_2, m, σ), where σ is a signature under an identity ID_1 and m is a message, ID_2 is an identity and $rk_{ID1\rightarrow ID2}$ is a re-signature key, output a re-signed signature σ^* under the identity ID_2.

Verify: Input a signature σ, check that if σ is a valid signature. If it holds then output 1, otherwise output 0.

IV. Analysis of Hu et al.'s Scheme

A. Review of the Scheme

Below is given the ***Setup***, ***Extract*** and ***ReKeyGen*** algorithms from Hu et al's scheme. The entire scheme is not provided due to space constraints, but is available in [5].

Let G_1 and G_2 be cyclic groups having the same prime order p, and e be a cryptographic bilinear map: $G_1 \times G_1 \rightarrow G_2$. The message space is G_2, and the identity space is Z_p.

Setup: The PKG randomly chooses generators g, $h \in G_1$ and $\alpha \in_R Z_p$, and sets $g_1 = g^{\alpha}$ and $mk = \alpha$. And randomly picks $h_1 \in G_1$ and set $u_1 = h_1^{\alpha}$.The public parameters are $params = \{g, g_1, h, h_1, u_1\}$, and the master secret key is α.

Extract: Let *ID* be the identity for which the private key is required. Choose r_{ID} randomly from Z_p, and compute
$h_{ID} = (hg^{-rID})^{1/(\alpha-ID)}$.
$d_{ID} = (r_{ID}, h_{ID})$ is the private key for the identity *ID*.

ReKeyGen: On input two private keys $d_{ID1} = (r_{ID1}, h_{ID1})$ and $d_{ID2} = (r_{ID2}, h_{ID2})$ corresponding to the identities ID_1 and ID_2 respectively, to generate a re-signature key $rk_{ID1\leftrightarrow ID2}$ from $ID_1 \leftrightarrow ID_2$, computes $rk_{ID1\leftrightarrow ID2} = (rk^{(A)}_{ID1\leftrightarrow ID2}, rk^{(B)}_{ID1\leftrightarrow ID2}) = (h_{ID2}/h_{ID1}, r_{ID2}/r_{ID1})$.

B. Attack on the Scheme

The above *ReKeyGen* algorithm violates the *Delegator Security* and *Delegatee Security* notions defined in the Introduction. If we take the first case, let us assume that the delegatee (ID_1) is honest, and that the proxy and the delegator (ID_2) are colluding.

The proxy has the ReKey value $rk_{ID1\leftrightarrow ID2} = (rk^{(A)}_{ID1\leftrightarrow ID2}, rk^{(B)}_{ID1\leftrightarrow ID2})$, and the delegator has its own private key $d_{ID2} = (r_{ID2}, h_{ID2})$. By using these values they can collude and obtain the private key of the delegatee, as follows:

$= (h_{ID2}/ rk^{(A)}_{ID1\leftrightarrow ID2}) = (h_{ID2}/ (h_{ID2}/h_{ID1})) = h_{ID1}$

$= (r_{ID2}/ rk^{(B)}_{ID1\leftrightarrow ID2}) = (r_{ID2}/ (r_{ID2}/r_{ID1})) = r_{ID1}$

Thus, they can obtain $(r_{ID1}, h_{ID1}) = d_{ID1}$. Using this, they can now produce signatures from the delegatee on any message.

In the second case, let us assume that the delegator (ID_2) is honest, and that the proxy and the delegatee (ID_1) are colluding. By performing the reverse of the steps given above, the proxy and delegatee will be able to jointly obtain the delegator's private key and use it to produce signatures on any message.

V. The Proposed ID-Based Proxy Re-signature Scheme

The new unidirectional Identity based proxy re-signature scheme, between a delegatee having identity A and delegator having identity B, is described below. It ensures that delegator and delegatee security are provided by inducing randomness in the proxy's rekey value.

Setup:

Let G_1 and G_2 be cyclic additive and multiplicative groups respectively having the same prime order p, and e be a cryptographic bilinear map: $G_1 \times G_1 \rightarrow G_2$. Let P be a generator of the group G_1. Randomly choose an $s \in Zq$ as the master secret, and define the corresponding master public value as $P_{pub}= s\cdot P$. Define two hash functions. H_1: $\{0, 1\}^* \rightarrow G_1$ and H_2: $\{0, 1\}^* \rightarrow Z_q$. The public parameters are $params = \{G_1, G_2, P, P_{pub}, H_1, H_2\}$, and the master secret key is s.

Key Extract:

For any user with identity *ID*, compute the public and private keys as follows:

- Public key : $Q_{ID} = H_1(_{ID})$
- Private key: $S_{ID} = s\cdot Q_{ID}$

Re-Key Gen:

On input the private keys S_A, S_B of the delegatee and delegator respectively, having identities A and B, the re-signature key for the proxy is computed in an interactive fashion as follows:

- Randomly choose some $r \in Z_q$
- $R_1 = S_B + (r\text{-}1)S_A$
- $R_2 = r\cdot Q_A$

Only $< R_1 , R_2 >$ are given to the proxy

Signature by Delegatee A:

The delegatee generates the signature as follows, using his private key:

- Randomly chooses some $R \in Z_q$
- Generates $\sigma_1 = H_2(m) \cdot S_A + R$, and $\sigma_2 = e(R,P)$

The signature σ on the message m is $<\sigma_1, \sigma_2>$

Verification of delegatee's signature:

The verifier checks if the following condition holds good:

$$e(\sigma_1,P) = e(H_2(m) \cdot Q_A,P_{pub}) \cdot \sigma_2$$

Re-Signature by Proxy:

On input a signature $\sigma = <\sigma_1, \sigma_2>$ on the message m, the proxy does the following:

- $\sigma_{11} = \sigma_1 + H_2(m) \cdot R_1$
- $\sigma_{12} = \sigma_2$
- $\sigma_{13} = R_2$

The transformed signature on m is $\sigma^* = <\sigma_{11}, \sigma_{12}, \sigma_{13}>$

Verification of the re-signature:

The verifier checks if the following condition holds good:

$$e(\sigma_{11},P) = e(H_2(m) \cdot (Q_B+\sigma_{13}),P_{pub}) \cdot \sigma_{12}$$

VI. Computational Complexity

The proposed scheme is computationally more efficient than the scheme in [3]. The expensive computations required in Hu's scheme are:

- Bilinear pairings: 4
- Point exponents: 10
- Point multiplication/division: 11

The new scheme requires the following expensive computations:

- Bilinear pairings: 2
- Point multiplication: 6
- Point addition/subtraction: 4
- Hash Computations: 4

As can be seen, the proposed scheme is computationally less intensive than Hu's existing scheme, while ensuring a higher degree of security.

VII. Security Analysis

A. Correctness

This proof is provided by proving the validity of the verification algorithm.

$L.H.S= e(\sigma_{11}, P)$

$=e(\sigma_1 + H_2(m) \cdot R_1, P)$

$=e(H_2(m) \cdot S_A + R+ H_2(m) \cdot (S_B + (r-1) S_A), P)$

$=e(H_2(m) \cdot (S_B + rS_A) +R, P)$

$=e(H_2(m) \cdot (Q_B + rQ_A), P_{pub}).e(R, P)$

$=e(H_2(m) \cdot (Q_B+\sigma_{13}), P_{pub}) \cdot \sigma_{12}$

$= R.H.S$

B. Unforgeability

As per the security notions, the proxy signature should not be susceptible to forgery in three scenarios:

Limited Proxy:

When both the delegator and delegate are honest, the proxy only knows the rekey values, i.e.:

$R_1 = S_B + (r-1) S_A$ and $R_2 = r \cdot Q_A$

Given these, obtaining the private keys of A and B can be reduced to the discrete log problem.

Let us also assume that the proxy has access to a number of messages and valid signatures by A on those messages. In that case, producing a valid signature on a new message m* can trivially be reduced to the CDH hard problem. The inputs to the adversary will be $<P,\ ap=Q_A \text{ and } bP=P_{pub}>$, and the output should be a part of the signature, i.e. it must include A's private key. $<abP = S_A>$.

Delegator and Delegatee Security:

These two cases are similar to the limited proxy notion. The only difference here will be the added value of one private key. However, since there is an added random value as part of the proxy's re-key, it becomes impossible for dishonest colluders to obtain the private key of the honest party. This problem can again be reduced to the discrete log problem.

The unforgeabilty of the signature of the delegate can be shown in the same way as described above. In case of delegator security, the CDH problem inputs are altered to $<P,\ ap = Q_B \text{ and } bP=P_{pub}>$, with output $<abP=S_B>$.

VIII. Inter-Domain Applications

In an inter-domain application, there exist two KGCs each maintaining an independent identity based system. The delegatee (A) is managed by KGC1 and the delegator (B) by KGC2. The following changes are to be made to the described proxy re-signature:

Setup:

The same algorithm described in section V. is used by each of the KGCs. Their generators are P_1, P_2 respectively. Their corresponding master secret and master public values are s_1, s_2 and $P_{pub1}= s_1 \cdot P_1$, $P_{pub2}= s_2 \cdot P_2$.They both use the same hash functions, i.e. $H_1 : \{0,1\}^* \rightarrow G_1$ and $H_2: \{0,1\}^* \rightarrow Z_q$. The public parameters are $params_1$ and $params_2$.

Key Extract:

For any user with identity ID, each KGC computes the public and private keys using their respective secret values.

Re-Key Gen:

On input the private keys S_A, S_B of the delegatee and delegator respectively, having identities A and B, the re-signature key for the proxy is computed in an interactive fashion as follows:

- Randomly choose some $r \in Z_q$
- $R_1 = S_B + (r-1)S_A$
- $R_2 = r \cdot Q_A$
- $R_3 = s_2P_1$
- Only $< R_1 , R_2 ,R_3>$ are given to the proxy

Signature by Delegatee A:

The delegatee generates the signature as follows, using his private key:

- Randomly chooses some $R \in Z_q$
- Generates $\sigma_1 = H_2(m) \cdot S_A + R$, and $\sigma_2 = e(R,P)$

The signature σ on the message m is $<\sigma_1, \sigma_2>$

Verification of delegatee's signature:

The verifier checks if the following condition holds good:

$$e(\sigma_1,P) = e(H_2(m) \cdot Q_A, P_{pub}) \cdot \sigma_2$$

Re-Signature by Proxy:

On input a signature $\sigma = <\sigma_1,\sigma_2>$ on the message m, the proxy does the following:

- $\sigma_{11} = \sigma_1 + H_2(m) \cdot R_1$
- $\sigma_{12} = \sigma_2$

- $\sigma_{13} = R_2$
- $\sigma_{14}=R_3$

The transformed signature on *m* is $\sigma^* = <\sigma_{11}, \sigma_{12}, \sigma_{13}, \sigma_{14}>$

Verification of the re-signature:

The verifier checks if the following conditions hold good:

$e(\sigma_{11},P1) = e(H_2(m)\cdot Q_B, \sigma_{14}) \cdot e(H_2(m)\cdot\sigma_{13},P_{pub1}) \cdot\sigma_{12}$

And,

$e(\sigma_{41},P2) = e(P1,Ppub2)$

IX. Conclusion

Thus, an attack on Hu et al's scheme is described, and a new ID Based Proxy Re-signature scheme is proposed, which is not susceptible to this attack. In terms of efficiency, this scheme is efficient, as it requires only two of the computationally expensive bilinear pairings, in comparison to the four pairing computations required in Hu et al's [5] scheme. The application of a modified version of this scheme can be used to ensure inter-domain security, as detailed. Inter-domain security has enormous potential in real-world applications, and the signature is one of the first attempts made to cater to this field.

Acknowledgment

I thank Dr. N.Vijayarangan, Senior Scientist of Tata Consultancy Services (TCS) Ltd, for encouraging me to work in the fields of Digital signatures and Inter-domain security. I am supported by them for exploring the applications of this research.

References

[1] A. Shamir, "Identity-based cryptosystems and signature schemes," In Crypto'84, LNCS 196, Springer-Verlag, pp.47-53, 1984.

[2] M. Blaze, G. Bleumer, and M.Strauss, "Divertible protocols and atomic proxy cryptography," In EUROCRYPT'98, LNCS 1403, Springer-Verlag, pp. 127-144, 1998.

[3] G. Ateniese and S. Hohenberger, "Proxy re-signatures: new definitions, algorithms, and applications," In ACM CCS'05, pp. 310-319, 2005.

[4] J. Shao, Z. Cao, L. Wang, and X. Liang, "Proxy re-signature schemes without random oracles," In INDOCRYP'07, LNCS 4859, Springer-Verlag, pp.197-209, 2007.

[5] X. Hu, Z. Zhang, and Y. Yang, "Identity Based Proxy Re-Signature Schemes without Random Oracle," In CIS '09, pp. 256-259, 2010.

[6] S. Chow and R. Phan, "Proxy Re-signatures in the Standard Model," In ISC '08, LNCS 5222, Springer-Verlag, pp.260-276, 2008.

[7] Q. Tang, P. Hartel, and W. Jonker, "Inter-Domain Identity Based Proxy Re-encryption," In *Information Security and Cyptology*, LNCS 5487, Springer-Verlag, pp. 332-347, 2009.