Identity-Based Redactable Lamport Signature Scheme

Fang Zuo¹, Qianli Wang², Pu Cheng*¹
¹Institute of Intelligent Network System, Henan University, Kaifeng, China
²School of Software, Henan University, Kaifeng, China
*chengpul@163.com

Abstract: The Redactable Signature Scheme (RSS) is one of the hottest topics in cryptography in these years. We find that even if the revision of the source document is limited to a very small range, the revision still has a high practical value. Therefore, it is necessary to find a solution that allows the trusted user to make the above restricted modifications to the source document. What's more, anyone who needs to verify the security of the modifications can do this. In order to solve this practical problem, we tried to introduce identities in the Lamport signature scheme to establish associations between keys and people, so that let Alice who was the owner of a document and has signed document then can authorize Bob to modify the document and generate new valid signatures. After discussion, we proved that if the identity system we use is secure, then our redactable signature scheme is also secure.

1. Introduction
Since the first public key algorithm was proposed by Diffie and Hellman [1] in 1976, as an important public key cryptography technology, digital signature verifies the integrity of data content and data source [2]. As human society has entered the data era, the security of data has become a major issue in relation to national economy, politics, national defence and cultural security [3]. Digital signature plays an irreplaceable role in guaranteeing data security.

Consider the following scenario: Alice gives Bob a cheque with blank amount. And only Bob is allowed to fill in the amount. So that anyone except Bob cannot cash this cheque. In addition to filling in the amount, filling in any other information may invalidate this cheque.

By the above example, we hope to show that, sometimes Alice, Signed a message $m$ and sent it with the signature $s$ to Bob, allow Bob to append some additional information to $m$ then get another message $m'$. To verify that the message $m'$ is valid, Bob should have the ability to generate a valid signature $s'$ for $m'$ without Alice's help in her absence. And if any others, except Alice and Bob, wants to use the message $m$ or $m'$, both $s$ and $s'$ are invalid.

In the next sections, we'll introduce our solution to the above problems. Before we begin our introduction of the solution, we must first introduce some of the previous studies that we will use.

2. Concepts

2.1 Redactable Signature
The concept of Redactable Signature Scheme (RSS) has been put forward successively by Rivest [4] and Johnson et al [5]. However Derler [6] et al. presents a general framework for redactable signatures for the first time in 2015, thus we use their definitions about redactable signatures.
A RSS allows any party to remove parts of a signed message such that the corresponding signature $s$ can be updated without the signers’ secret key $x$ (A sample message is shown in Figure 1. By the way, in Figure 1, the middle part shows that how to edit the message in left part to get another meaningful message. Note that the intact meaning of the message in left part has been covered up by the message in middle part. So it is necessary to disclose explicitly of modification for message, just like the right part in Figure 1.) The so derived signature $s'$ then still verifies under the signers' public key $y$.

Definition 1: If there is an abstract data structure used to describe the rules of redactions including descriptions of dependencies, fixed elements or relationship between elements, it is named ADM. A specific description of how the message $M$ should be redacted is called the MOD. If we let MOD is a function that only output 0 or 1, and function MOD takes two parameters: ADM and $M$. Then MOD output 1 means that MOD is a valid redaction description applies to ADM and $M$. If we let ADM is a function that only output 0 or 1. When $M$ is the only parameter of function ADM, if function $ADM = 1$ means that ADM can match $M$, in the other words, ADM is valid for $M$.

Definition 2: An RSS consists of four efficient algorithms: a key generation algorithm KG, a signature algorithm Sig, a verification algorithm Verify and a redaction algorithm Red, which are defined as follows:

KG: this algorithm needs to input a security parameter $\lambda$, and a key pair $(x, y)$ will be output by this probabilistic algorithm.

Sig: this algorithm needs three parameters: a secret key $x$, a message $M$ and ADM, a pair of message and signature $(M, s)$ together with some auxiliary redaction information $r$ will be returned by Sig as the result. Under ideal conditions, ADM should always be right and clearly derived from any valid pair of message and signature. Another point need to focus on that ADM may change through a redaction.

Verify: this algorithm needs three parameters: a public key $y$, a signature $s$ and a message $M$, as the result of this deterministic algorithm, a value for 0 or 1 will be output.

Redact: This algorithm may be probabilistic, takes four parameters including a public key $y$, a message $M$ with its valid signature $s$, and an $r$ denotes MOD and auxiliary redaction information as input. Redact algorithm returns a pair $(M', s')$ of redacted message-signature, and it also updates the auxiliary redaction information $r$. Note that the ADM may be altered by this algorithm in an unambiguous way.

A RSS which applies the above specifications also requires that Sig must return an error value, when ADM = 0, while Redact must also return an error value, when MOD = 0. Note that if no auxiliary redaction information is required, $r$ can also be $\emptyset$.

![Figure 1 Illustration of the redactable message](image)

Alice falls in love with Bob, who is a handsome boy.  
Alice falls in love with a handsome boy.  
Alice falls in love with **** ***** a handsome boy.
2.2 Identity-based Signature

In Shamir's paper [7], he introduced an encryption scheme that enables any pair of users to communicate securely and verify each other's signatures without the need to exchange private or public keys, also without the need to retain key directories or use a third party service. The scheme assumes there is a trusted centre whose sole purpose is to generate keys for users by providing personalized smart cards for each user when they first join the network. The information embedded in the card enables the user to sign and encrypt the message he sent, and decrypt and verify the message he receives in a completely independent way, regardless of the identity of the other users in the same network. When a new user joins the network, previously issued cards do not have to be updated, and the centres do not have to coordinate their activities or even remember the user list. After all cards are issued, the centre can be closed and the network can continue to work properly in an indefinitely completely decentralized way. Shamir's identity-based signature scheme is shown in Figure 2.

\[\text{Figure 2 Architecture of identity-based signature scheme}\]

3. Redactable Lamport Signature Scheme

3.1 Lamport Signature

Lamport Signature [8] was invented by Leslie Lamport. Next, we will briefly introduce Lamport Signature.

3.1.1 Assumption: assume D is the set of possible documents, C is the set of possible keys, and possible values make up a set E. Assume that there is a set of 2n consecutive positive integer elements from 1 to 2n, we may call this set as T. If a set is a subset of T and it has exactly n elements, we may call this T's subset as t. And all of the possible t makes up a set \(\Sigma\). And then assume there are two functions U and V, where the function U creates a mapping from C to E, function V creates a mapping from D to \(\Sigma\).

Function U has the following two properties:
For each value \( v \) in \( E \), it is infeasible to find a key \( k \) in \( C \) such that \( U(k) = v \) by computing.

If \( E \) is a subset of \( E \) and the size of \( E \) is enough small, it is easy to find a key \( k \) such that \( U(k) \) is not equal to any element of \( E \).

Function \( V \) has the following property:

Given any document \( m \) in \( D \), and there is a document \( m' \) in \( D \). If \( m' \neq m \), then \( V(m') \neq V(m) \), and if \( V(m') \neq V(m) \) then \( m' \neq m \).

### 3.1.2 The Signature Process:
Alice, who as the sender, first picks \( 2n \) keys \( k_i \) such that all the values \( U(k_i) \) are distinct. She puts these \( 2n \) values into a public repository as the data item \( \alpha = (U(k_1), \ldots, U(k_{2n})) \).

Then Alice computes \( V(m) \) to get an element of \( \Sigma \), in other words, Alice has \( n \) positive integers. It also is required that those \( n \) positive integers order by ascending. Alice uses these \( n \) integers as indexes to pick up \( n \) keys \( k_i \), and she uses these \( k_i \) to make up a signature for document \( m \). So that Alice has generated a signature \( s \).

After Alice obtained the signature \( s \), she can dispose all traces of the \( n \) keys \( k_q \) with \( q \) not in \( V(m) \).

### 3.1.3 The Verification Process:
To verify that a \( n \)-tuple \((h_1, \ldots, h_n)\) is a valid signature for the document \( m \), Victor first computes \( V(m) \) to obtain the \( n \) indexes \( j \) that order by ascending. And then Victor picks out \( n \) values \((v_1, \ldots, v_n)\) that pointed by \( j \) from \( \alpha \). If each \( v_i \) can match corresponding \( h_i \) by computing \( U(h_i) = v_i \), then for document \( m \), this \( n \)-tuple \((h_1, \ldots, h_n)\) is a valid signature.

### 3.2 The Idea
We notice that Lamport has pointed out and proved in his literature [8] that his signature scheme has the following two properties:

Property a: If Peter does not reveal any of the keys \( k_i \), then Queenie cannot generate a valid signature \( s \) for any document \( m \).

Property b: If any of the keys \( k_i \) except the ones that are contained in the signature \( s \) does not be revealed by Peter, then Queenie cannot generate a signature \( s' \) to make \( s' \) valid for any document \( m' \) with \( m' \neq m \).

For the above second property, in other words, we propose the following property:

Property c: If Peter reveals any of the keys \( k_i \) which is not contained in the signature \( s \), then Queenie can generate a valid signature \( s' \) for another document \( m' \) with \( m' \neq m \).

Base on this property, we guess that we can construct a redefinable signature scheme by expose keys discretely. But before we construct such a signature scheme, we should first prove property c.

Proof 1: If Peter reveals a key \( k_q \) which is not contained in the signature \( s \), so that Queenie will get \( n+1 \) keys \( k \) including \( n \) \( k_q \) and \( k_p \). Then Queenie can get each index of these keys by computing \( U(k) \) and matching to \( E \). No matter how small the probably is, Queenie can get a set of \( n \) indexes includes \( n-1 \) \( k_q \) and \( k_p \) by computing function \( V \). Thus, Queenie can generate a valid signature \( s' \) for document \( m' \). And there is always \( m' \neq m \), because of property of function \( V \).

### 3.3 The implement

#### 3.3.1 Assumption:
As same as Lamport Signature, we assume there are \( D, C, E, \Sigma \) have the same meaning as section 3.1.1. We also assume there are two one-way functions \( U \) and \( V \), as same as Lamport’s one-way functions, both have the same properties.

#### 3.3.2 The Signature Process:
Alice and Bob will generate signature by the following steps:

Step 1: Alice chooses \( 2n \) keys \( k_i \) such that all the values \( v_i = U(k_i) \) are distinct. And she puts these \( 2n \) values in a private repository as a data item \( \alpha \) so that \( \alpha = (U(k_1), \ldots, U(k_{2n})) \).
Step 2: Alice computes $V(m)$ so that she can obtain a set of $n$ integers $\{i_1, \ldots, i_m\}$. The signature $s$, for document $m$, consists of the $n$ keys $k_j$. Where the $j$ are defined by the following two requirements:

1. $j$ is one element of $V(m) = \{i_1, \ldots, i_m\}$.
2. The elements of $V(m)$ order by ascending.

Step 3: Alice discards the $n$ keys which are not contained in $s$, and destroys corresponding values from $a$.

Step 4: Alice gets another $n$ values $v_j$ from Bob but does not know the keys $k_j$ of $v_j$. What's more, Alice adds these values $v_j$ to $a$ and keeps the $n$ values $v_j$ from herself in positions where they are before step 3.

Step 5: Alice publishes $a$.

Note we define step 4 as authorization.

3.3.3 The Redact Signature Process: Bob gets document $m$ and its signature $s$ from Alice. Notice that Bob now knows all $2n$ keys of signature $s$. Then a valid signature $s'$ for $m'$, where $m'$ are another document and its source is document $m$, can be generated by Bob through the following steps:

Step 6: Bob can freely add, but only add, any content to document $m$ to make $m$ becomes another document $m'$. It means that $m$ must be a part of $m'$.

Step 7: Then Bob can execute the process which described in section 3.1.2 to generate signature $s^*$ for document $m^*$.

So far, we have two documents $m$ and $m^*$, and we also have their corresponding signatures $s$ and $s^*$.

3.3.4 The Verification Process: It is easy to independently verify two documents and their respective signatures. But the extra condition is that if and only if document $m$ can be intercepted from document $m^*$, document $m^*$ and its signature $s^*$ are valid.

3.4 Security

It is clear that our scheme is not secure at present. Because:

1. Everyone who had seen $m$ and $s'$ know the superset of keys that make up signature $s$. And we have already proved that they can forge a valid signature.

2. More serious is that if Bob is dishonest, he can forge the source document $m$ at will. This is obvious when Bob knows all the keys.

For the above reasons, we introduce the identity system to improve the security of the scheme. We only propose the basic assumptions about identity system, once there is enough property of identity system we assumed can be used in our scheme, we can immediately re-discuss the security about our RSS.

We assume that the keys and identity are related. For an example (shows in Figure 3), we may regard this relationship as that the keys contain identity information. Because of there is authorized, keys that make up $s$ or $s'$ only contain Alice's and Bob's identity information. Thus if another one except Alice and Bob holds $s$ or $s'$, it will be considered invalid. This solves the previous first unsecure point.

Because the keys that make up the signature $s$ of document $m$ are all from Alice. Based on the property of one-way function $V$, there is no second combination of keys that makes these keys which make up signature $s'$ for document $m'$, with $m' \neq m$, contains and only contains Alice's identity information. In this way, Bob cannot forge the source document $m$. Then all the additions to document $m$, operated by Bob, are allowed. This solves the previous second unsecure point.

Now, if the identity system we using is safe, then our scheme is safe. Next, we will prove this conclusion.

Proof 2: If the identity system is safe, means that no-one else can pretend to be Alice and Bob. So that if signature $s$ is generated by Alice and signature $s'$ is generated by Bob, the third person cannot use the both signature because his identity information is not contained in keys. And Bob cannot forge source document $m$ so that $m$ and $s$ are safe and valid, then $m'$ and $s'$ are valid too.
In addition, if Bob obeys the limit of only content that is allowed to addition, our redactable signature scheme is actually feasible.

![A valid signature scheme](image)

**Figure 3** Signature contain identity

### 4. Conclusion

We've already described how we apply the identity system to Lamport Signature Scheme to make it redactable. And we also discussed the security of our scheme.

Our scheme is still essentially a Lamport Signature Scheme, so our scheme has the same flaws and advantages as Lamport Signature Scheme. And fortunately, the improvements to Lamport Signature Scheme, like Merkle's [9], also apply to ours.

Most of the redactable digital signatures are not efficient enough [10]. Thus how to improve efficiency of RSS will be one of the important points of our next research.

### References

[1] Diffie, W., and Hellman, M.: 'New directions in cryptography', IEEE transactions on Information Theory, 1976, 22, (6), pp. 644-654

[2] Katz, J., and Lindell, Y.: 'Introduction to modern cryptography' (CRC press, 2014)

[3] Brown, B., Chui, M., and Manyika, J.: 'Are you ready for the era of 'big data", McKinsey Quarterly, 2011, 4, (1), pp. 24-35

[4] Rivest, R., 'Two new signature schemes', (Cambridge seminar, 2001)

[5] Johnson, R., Molnar, D., Song, D., et al.: 'Homomorphic signature schemes'. Cryptographers' Track at the RSA Conference. Springer, Berlin, Heidelberg, 2002, pp. 244-262

[6] Derler, D., Pöhls, H.C., Samelin, K., et al.: 'A general framework for redactable signatures and new constructions'. International Conference on Information Security and Cryptology. Springer, Cham, 2015, pp. 3-19

[7] Shamir, A.: 'Identity-based cryptosystems and signature schemes'. Workshop on the theory and application of cryptographic techniques. Springer, Berlin, Heidelberg, 1984, pp. 47-53

[8] Lamport, L.: 'Constructing digital signatures from a one-way function', Vol. 238, Palo Alto: Technical Report CSL-98, SRI International, 1979

[9] Merkle, R.C.: 'A certified digital signature'. Conference on the Theory and Application of Cryptology. Springer, New York, NY, 1989, pp. 218-238.

[10] Ma, J., Liu, J., Wang, M., et al.: 'An efficient and secure design of redactable signature scheme with redaction condition control'. International Conference on Green, Pervasive, and Cloud Computing. Springer, Cham, May, 2017, pp. 38-52