A Note On Libert-Vergnaud Proxy Re-encryption Scheme

Zhengjun Cao*

Abstract In 2008, Libert and Vergnaud put forth a proxy re-encryption scheme (PRE). Unlike some earlier PRE schemes, the LV08 scheme specifies a validity-checking process to guarantee that the received ciphertext is well-formed. In this note, we clarify an ultimate fact that *the received message is well-formed is the premise to decryption in all encryption schemes.* The underlying mechanism to keep the communicated message well-formed in encryption schemes is another topic. The authors ignored the fact and proposed a cumbersome presentation. We will simplify the LV08 scheme and show its security level is the same as that of the scheme proposed by Ateniese et al in 2005. Therefore, the LV08 scheme can not ensure chosen-ciphertext security as claimed.

Keywords proxy re-encryption, bilinear map, single-hop

1 Introduction

Proxy re-encryption (PRE) allows a proxy to turn a ciphertext computed under Alice's public key into one that can be opened by Bob's secret key. It is a special kind of proxy encryption schemes where delegatees only need to store their own decryption key. It has many applications, such as email forwarding, law enforcement, and performing cryptographic operations on storage-limited devices. Especially, in the scenario that the delegator has less computational power to re-encrypt the received message for the delegatee. The delegator can directly forward the message to a semi-trusted proxy and ask the proxy to re-encrypt it.

In 1998, Blaze et al [2] introduced proxy re-encryption, where a proxy is given a re-encryption key that allows it to turn a message encrypted under public key pk_1 into an encryption of the same message under a different public key pk_2 without being able to learn anything about the encrypted message. There are two types of proxy re-encryption schemes. If the re-encryption key $rk_{1,2}$ necessarily allows the proxy to turn ciphertexts under pk_1 into ciphertexts under pk_2 and vice versa, then the scheme is called bidirectional. If the re-encryption key $rk_{1,2}$ allows the proxy to turn only from pk_1 to pk_2, then the scheme is called unidirectional.

In 2005, Ateniese et al [1] presented a unidirectional PRE scheme. But the re-encryption

[1]Department of Mathematics, Shanghai University, China. caozhj@yahoo.cn

algorithm is single-hop, that is, a re-encrypted ciphertext cannot be further re-encrypted. In contrast, the BBS98 scheme is multi-hop, namely a ciphertext can be re-encrypted from Alice to Bob to Carol and so on. In 2007, Canetti and Hohenberger [4] presented a multi-hop PRE scheme and the definition of security against chosen-ciphertext attacks for PRE schemes.

In 2008, Libert and Vergnaud [7] put forth a proxy re-encryption scheme (LV08 for short). Unlike some earlier PRE schemes, the LV08 scheme specifies the validity-checking process to guarantee that the received ciphertext is well-formed. In this paper, we clarify an ultimate fact that *the received message is well-formed is the premise to decryption in all encryption schemes*. The underlying mechanism to keep the communicated message well-formed in encryption schemes is another topic. The authors ignored the fact and proposed a cumbersome presentation. We will simplify the LV08 scheme and show its security level is the same as that of the scheme proposed by Ateniese et al in 2005. Therefore, the LV08 scheme can not ensure chosen-ciphertext security as claimed.

2 Blaze-Bleumer-Strauss proxy re-encryption scheme

In 1998, Blaze et al [2] proposed the first proxy re-encryption scheme (BBS98 for short). It is based on the ElGamal encryption [6]. The scheme can be described as follows.

Setup Let $(\mathbb{G}, \cdot)$ be a group of prime order p and g be a generator of $\mathbb{G}$.

KeyGen The delegator $\mathcal{U}_i$ and the delegatee $\mathcal{U}_j$ set their public keys as $X_i = g^{x_i}, X_j = g^{x_j}$ for random $x_i, x_j \overset{R}{\leftarrow} \mathbb{Z}_p^*$, separately.

ReKeyGen Generate the re-encryption key $R_{ij} = x_j/x_i$, which is given to the proxy $\mathcal{P}$.

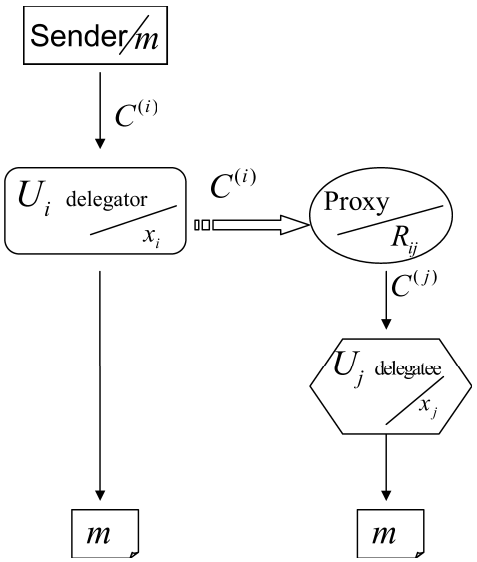
Encrypt To encrypt a message $m \in \mathbb{G}$ under the public key X_i, the sender picks $r \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and computes $\alpha = X_i^r, \quad \beta = g^r \cdot m$. The ciphertext is $C^{(i)} = (\alpha, \beta)$.

ReEncrypt Given a ciphertext $C^{(i)} = (\alpha, \beta)$ intended for $\mathcal{U}_i$, the proxy $\mathcal{P}$ with the key $R_{ij} = x_j/x_i$ turns it into a ciphertext intended for $\mathcal{U}_j$ by computing $\gamma = \alpha^{R_{ij}}$. The re-encrypted ciphertext is $C^{(j)} = (\gamma, \beta)$.

Decrypt To decrypt a ciphertext $C^{(i)}, \mathcal{U}_i$ computes $m = \beta/\alpha^{1/x_i}$. To decrypt a re-encrypted ciphertext $C^{(j)}, \mathcal{U}_j$ computes $m = \beta/\gamma^{1/x_j}$.

To capture the nature of the cryptographic primitive, we refer to the Graph 1.

Limitations of BBS98. Blaze et al [2] point out an inherent limitation in the BBS98 scheme: the proxy key x_j/x_i also allows translating ciphertexts from $\mathcal{U}_j$ to $\mathcal{U}_i$, which may be undesirable in some situations. There is another shortcoming, namely the proxy and the delegatee $\mathcal{U}_j$ can collude to expose the delegator's private key x_i given x_j/x_i and x_j.



Graph 1: Proxy re-encryption model

3 Ateniese-Fu-Green-Hohenberger proxy re-encryption scheme

In 2005, Ateniese et al proposed the first unidirectional proxy re-encryption scheme (AFGH05 for short) using bilinear maps [3]. Groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p are called bilinear map groups if there is a mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties: 1) $e(g^a, h^b) = e(g, h)^{ab}$ for any $(g, h) \in \mathbb{G} \times \mathbb{G}$ and $a, b \in \mathbb{Z}$; 2) efficient computability for any input pair; 3) $e(g, h) \neq 1_{\mathbb{G}_T}$ whenever $g, h \neq 1_{\mathbb{G}}$. The AFGH05 scheme can be described as follows.

Setup Choose bilinear map groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Choose a generator $g \xleftarrow{R} \mathbb{G}$. The system parameters are $\mathbb{G}, \mathbb{G}_T, p, g, e$.

KeyGen The delegator $\mathcal{U}_i$ and the delegatee $\mathcal{U}_j$ set their public keys as $X_i = g^{x_i}, X_j = g^{x_j}$ for random $x_i, x_j \xleftarrow{R} \mathbb{Z}_p^*$, separately.

ReKeyGen Given $\mathcal{U}_j$'s public key X_j, the delegator $\mathcal{U}_i$ with private key x_i can generate the re-encryption key $R_{ij} = X_j^{1/x_i}$ for a proxy $\mathcal{P}$.

Encrypt To encrypt a message $m \in \mathbb{G}_T$ under the public key X_i, the sender picks $r \xleftarrow{R} \mathbb{Z}_p^*$ and computes $\alpha = X_i^r, \beta = e(g, g)^r \cdot m$. The ciphertext is $C^{(i)} = (\alpha, \beta)$.

ReEncrypt Given a ciphertext $C^{(i)} = (\alpha, \beta)$, the proxy $\mathcal{P}$ with the key $R_{ij} = g^{x_j/x_i}$ can turn it into a ciphertext intended for $\mathcal{U}_j$ by computing $\gamma = e(\alpha, R_{ij})$. The re-encrypted ciphertext is $C^{(j)} = (\gamma, \beta)$.

Decrypt To decrypt a ciphertext $C^{(i)}$, $\mathcal{U}_i$ computes $m = \beta / e(\alpha, g)^{1/x_i}$. To decrypt a re-encrypted ciphertext $C^{(j)}$, $\mathcal{U}_j$ computes $m = \beta / \gamma^{1/x_j}$.

4 Libert-Vergnaud proxy re-encryption scheme

In 2008, Libert and Vergnaud [7] proposed a unidirectional proxy re-encryption scheme (LV08 for short). The scheme can be described as follows.

Setup Given a security parameter λ, choose bilinear map groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order $p > 2\lambda$ with a bilinear map $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$. Choose generators $g, u, v \xleftarrow{R} \mathbb{G}$ and a strongly unforgeable one-time signature scheme $Sig = (\mathcal{G}, \mathcal{S}, \mathcal{V})$. The global parameters are $\mathbb{G}, \mathbb{G}_T, p, g, u, v, e, Sig$.

KeyGen The delegator $\mathcal{U}_i$ and the delegatee $\mathcal{U}_j$ set their public keys as $X_i = g^{x_i}, X_j = g^{x_j}$ for random $x_i, x_j \xleftarrow{R} \mathbb{Z}_p^*$, separately.

ReKeyGen Given $\mathcal{U}_j$'s public key X_j, the delegator $\mathcal{U}_i$ with private key x_i can generate the re-encryption key $R_{ij} = X_j^{1/x_i}$ for a proxy $\mathcal{P}$.

Encrypt To encrypt a message $m \in \mathbb{G}_T$ under the public key X_i, the sender conducts the following steps.

1. Select a one-time signature key pair $(ssk, svk) \xleftarrow{R} \mathbb{G}(\lambda)$ and set $C_1 = svk$.
2. Choose $r \xleftarrow{R} \mathbb{Z}_p^*$ and compute $\alpha = X_i^r, \beta = e(g, g)^r \cdot m, C_4 = (u^{svk} \cdot v)^r$.
3. Generate a one-time signature $\sigma = S(ssk, (\beta, C_4))$ on the pair (β, C_4).

The ciphertext is $C^{(i)} = (C_1, \alpha, \beta, C_4, \sigma)$.

ReEncrypt Given a ciphertext $C^{(i)}$, the proxy $\mathcal{P}$ checks the validity of the ciphertext by testing the following conditions

$$e(\alpha, u^{C_1} \cdot v) = e(X_i, C_4) \tag{1}$$

$$\mathcal{V}(C_1, \sigma, (\beta, C_4)) = 1 \tag{2}$$

If well-formed, the proxy $\mathcal{P}$ can turn it into a ciphertext intended for $\mathcal{U}_j$ by choosing $t \xleftarrow{R} \mathbb{Z}_p^*$ and computing $C_2' = X_i^t, C_2'' = R_{ij}^{1/t} = g^{(x_j/x_i)t^{-1}}, C_2''' = \alpha^t = X_i^{rt}$. The re-encrypted ciphertext is $C^{(j)} = (C_1, C_2', C_2'', C_2''', \beta, C_4, \sigma)$.

Decrypt To decrypt a ciphertext $C^{(i)}$, $\mathcal{U}_i$ checks its validity. If the relations (1)-(2) hold, $\mathcal{U}_i$ computes $m = \beta / e(\alpha, g)^{1/x_i}$.

To decrypt a re-encrypted ciphertext $C^{(j)}$, $\mathcal{U}_j$ checks its validity by testing

$$e(C_2', C_2'') = e(X_i, g) \tag{3}$$

$$e(C_2''', u^{C_1} \cdot v) = e(C_2', C_4) \tag{4}$$

$$V(C_1, \sigma, (\beta, C_4)) = 1 \tag{5}$$

If relations (3)-(5) hold, $\mathcal{U}_j$ computes $m = \beta / e(C_2'', C_2''')^{1/x_j}$.

5 Analysis of LV08 scheme

5.1 Analysis

Unlike BBS98 and AFGH05 schemes, the LV08 scheme specifies the process to publicly verify the ciphtertext, which can decide whom the ciphertext is intended for and check the validity of the received ciphertext. Concretely, to check that the received ciphertext is well-formed, the recipient of a re-encrypted ciphertext needs to know who the original receiver is.

Here we clarify an ultimate fact that *the received message is well-formed is the premise to decryption in all encryption schemes.* The underlying mechanism to keep the communicated message well-formed in encryption schemes is another topic. The authors [7] ignored the fact and proposed a cumbersome presentation, which might distract the reader from investigating the security and cost of the scheme. Incidently, the proxy can easily specify the original receiver by binding an additional descriptor with the re-encrypted ciphtertext and sign them.

Based on the observations, we now simplify the LV08 scheme. By the simplification (see Table 1), one can see the main differences between the LV08 scheme and the AFGH05 scheme.

	BBS98	AFGH05	LV08 (simplificd)
Setup	$\mathbb{G}, p, g$	$\mathbb{G}, \mathbb{G}_T, p, g, e$	$\mathbb{G}, \mathbb{G}_T, p, g$
KeyGen	$X_i = g^{x_i}, X_j = g^{x_j}$	$X_i = g^{x_i}, X_j = g^{x_j}$	$X_i = g^{x_i}, X_j = g^{x_j}$
ReKeyGen	$R_{ij} = x_j/x_i$	$R_{ij} = g^{x_j/x_i}$	$R_{ij} = g^{x_j/x_i}$
Encrypt	$\alpha = X_i^r, \beta = g^r \cdot m$ $C^{(i)} = (\alpha, \beta)$	$\alpha = X_i^r, \beta = e(g,g)^r \cdot m$ $C^{(i)} = (\alpha, \beta)$	$\alpha = X_i^r, \beta = e(g,g)^r \cdot m$ $C^{(i)} = (\alpha, \beta)$
ReEncrypt	$\gamma = \alpha^{R_{ij}}$ $C^{(j)} = (\gamma, \beta)$	$\gamma = e(\alpha, R_{ij})$ $C^{(j)} = (\gamma, \beta)$	$\gamma_1 = \alpha^t = X_i^{rt}$ $\gamma_2 = R_{ij}^{1/t} = g^{(x_j/x_i)t^{-1}}$ $C^{(j)} = (\gamma_1, \gamma_2, \beta)$
Decrypt	$\mathcal{U}_i: \ m = \beta/\alpha^{1/x_i}$ $\mathcal{U}_j: \ m = \beta/\gamma^{1/x_j}$	$\mathcal{U}_i: \ m = \beta/e(\alpha,g)^{1/x_i}$ $\mathcal{U}_j: \ m = \beta/\gamma^{1/x_j}$	$\mathcal{U}_i: \ m = \beta/e(\alpha,g)^{1/x_i}$ $\mathcal{U}_j: \ m = \beta/e(\gamma_1,\gamma_2)^{1/x_j}$
Security of delegator's key	No	Yes	Yes
Security level	chosen-plaintext	chosen-plaintext	chosen-plaintext

Table 1: Comparisons of three PRE schemes

Comparing the simplified LV08 scheme with AFGH05 scheme, it is easy to find the LV08 scheme splits γ (in the AFGH05 scheme) into tow components (γ_1, γ_2) such that $\gamma = e(\gamma_1, \gamma_2)$. Since β is still of the form $e(g,g)^r \cdot m$, we conclude that the security level of the ciphertext $(\gamma_1, \gamma_2, \beta)$ is the same as that of the ciphertext (γ, β). To see this point, it only needs to turn $(\gamma_1, \gamma_2, \beta)$ into $(e(\gamma_1, \gamma_2), \beta)$.

In view of that Libert and Vergnaud [7] claimed the AFGH05 scheme can only ensure chosen-plaintext security, we conclude the LV08 scheme does not ensure chosen-ciphertext security as claimed.

5.2 A concrete attack

Now we provide a concrete chosen-ciphertext attack against the LV08 scheme. For convenience, we only describe the attack against the encrypted message, not against the re-encrypted message.

Suppose that the adversary *traps* a ciphertext $C^{(i)} = (C_1, \alpha, \beta, C_4, \sigma)$. That means the ciphertext and its corresponding one-time signature keys is only recorded by the *encryption oracle*, and it does not been queried to the *decryption oracle* previously. Hence, the adversary can use the corresponding one-time signature keys to make a new signature. Practically speaking, it is usual that the decryption oracle can not been timely informed that the system parameter Sig is updated.

Note that the above trapping assumption does not violate the specification made by the authors [7] that one-time signature meets that no PPT adversary can create a new signature for a previously signed message. By the way, the mechanism of validity-checking used in the Cramer-Shoup encryption [5] does differ from that used in the LV08 scheme.

After trapping the ciphertext $C^{(i)} = (C_1, \alpha, \beta, C_4, \sigma)$, the adversary only needs to pick $\theta \overset{R}{\leftarrow} \mathbb{Z}_p^*$ and compute

$$\widetilde{\alpha} = \alpha^{\theta}, \widetilde{\beta} = \beta^{\theta}, \widetilde{C_4} = C_4^{\theta}$$

and generate a one-time signature $\widetilde{\sigma} = S(ssk, (\widetilde{\beta}, \widetilde{C_4}))$ on the pair $(\widetilde{\beta}, \widetilde{C_4})$. Then the adversary query the decryption oracle with $(C_1, \widetilde{\alpha}, \widetilde{\beta}, \widetilde{C_4}, \widetilde{\sigma})$. Apparently, $(C_1, \widetilde{\alpha}, \widetilde{\beta}, \widetilde{C_4}, \widetilde{\sigma})$ meets the following testings

$$e(\widetilde{\alpha}, u^{C_1} \cdot v) = e(X_i, \widetilde{C_4})$$

$$\mathcal{V}(C_1, \widetilde{\sigma}, (\widetilde{\beta}, \widetilde{C_4})) = 1$$

In fact,

$$e(\widetilde{\alpha}, u^{C_1} \cdot v) = e(\alpha^{\theta}, u^{C_1} \cdot v) = e(\alpha, u^{C_1} \cdot v)^{\theta} = e(X_i, C_4)^{\theta} = e(X_i, \widetilde{C_4})$$

Upon receiving the output of the decryption oracle, the adversary can obtain

$$m^{\theta} = \widetilde{\beta} / e(\widetilde{\alpha}, g)^{1/x_i}$$

and recover the message m from m^{θ} because the adversary knows θ^{-1}.

Remark 1 The mechanism of introducing the process to check the validity of the received ciphertext in LV08 scheme, is inspired by the CH07 scheme [4]. Removing the redundant validity-checking process in CH07 scheme, we find it is just the version of the BBS98 scheme using bilinear maps. Therefore, it does not ensure chosen-ciphertext security as claimed.

Remark 2 The PRE scheme in [8] is directly inspired by AFGH05 and LV08 schemes. It contains a validity-checking process like LV08 scheme but does not split γ in AFGH05 scheme. Removing the redundant process in the PRE scheme, it is easy to find that the scheme does not ensure chosen-ciphertext security as claimed, too.

6 Conclusion

In this paper, we clarify that it is unnecessary to contain a validity-checking process in an encryption scheme. Based on the observation, we analyze the LV08 scheme and show that it can not ensure the chosen-ciphertext security. We also point out some representational PRE schemes can not ensure chosen-ciphertext security as claimed.

References

[1] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage. In: proceedings of NDSS'05, The Internet Society, 29-43 (2005)

[2] M. Blaze, G. Bleumer, M. Strauss. Divertible protocols and atomic proxy cryptography. EUROCRYPT'98. Lecture Notes in Computer Science, Vol 1403, 127-144, Springer (1998)

[3] D. Boneh, M. Franklin. Identity-based encryption from the Weil Pairing. SIAM Journal of Computing, 32 (3):586-615 (2003)

[4] R. Canetti, S. Hohenberger, Chosen-ciphertext secure proxy re-encryption. ACM Conference on Computer and Communications Security, 185-194, ACM (2007)

[5] R. Cramer, V. Shoup. A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack, CRYPTO'98, LNCS 1462, pp. 13-25. Springer (1998)

[6] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms, In: *Advances in Cryptology CRYPTO'84*, Lectures Notes in Computer Science 196, pp. 10-18. Springer (1984)

[7] B. Libert, D. Vergnaud, Unidirectional chosen-ciphertext secure proxy re-encryption. Public Key Cryptography 2008. Lecture Notes in Computer Science, Vol 4939, 360-379, Springer (2008)

[8] J. Wen, MR Chen, YJ Yang, R. H. Deng, KF Chen, F. Bao, CCA-Secure Unidirectional Proxy Re-Encryption in the Adaptive Corruption Model without Random Oracles, Science China: Information Science, 53 (3), 593-606, Springer (2010)