LETTER

Comparing Performance of Hierarchical Identity-based Signature Schemes

Peixin CHEN[†a)], ***Member***, **Yilun WU**[†], **Jinshu SU**[†*], ***and*** **Xiaofeng WANG**[†], ***Nonmembers***

SUMMARY Key escrow problem and high computational cost are the two major problems that hinder the widely adoption of hierarchical identity-based signature (HIBS) scheme. HIBS schemes with either escrow-free (*EF*) or online/offline (*OO*) model are proposed and proved secure by Chen et al. However, they only focus on the formal proof of security and have not yet evaluated their schemes either theoretically or experimentally. In this letter, several EF/OO HIBS schemes are considered. We study the algorithmic complexity of the schemes, and discuss the scheme performance and practicability of *EF* and *OO* models.

key words: *identity-based signature, computational complexity, paring based cryptography*

1. Introduction

Identity based signature (IBS) scheme is a digital signature scheme which allows a receiver to verify message using the signer's identity as public key [1]. Utilizing private key generator (PKG) to generate private key for users, Choon et al. propose the first practical IBS scheme [2]. Gentry et al. then present the first hierarchical IBS (HIBS) scheme, which imposes domain PKGs to reduce the workload on root PKG and solve the single-point failure problem in IBS scheme [3].

Since the user private keys are generated by PKGs, a PKG knows the keys so that it can sign messages unscrupulously without being detected. Such problem is referred to as key escrow problem. Escrow-free HIBS schemes have been proposed to address the problem. Chen et al. propose an escrow-free model that can extent any primitive HIBS scheme to solve the key escrow problem [4]. They applied the model to the SHER-IBS scheme [5], obtaining a secure scheme CWS-EF-HIBS.

Besides the key escrow problem, the low computation efficiency of HIBS scheme is another concern while deploying the identity-based signature scheme. Most IBS schemes involve computations including pairings over points on eliptic cure and point multiplications in groups, which might be too costly to be applied in lightweight devices. Online/offline (OO) signature mechanism that divides the process of message signing into offline phase and online phase is an effective method to reduce the computational cost of signature generation [6]. Imposing the *OO* mechanism, numerous identity-based online/offline signature (IBOOS) scheme have been proposed [7]–[9]. By extending the SHER-IBS scheme, Chen et al. propose a hierarchical IBOOS (HIBOOS) scheme with high online signing efficiency [8]. On this basis, Chen et al. propose a user-selected secret model and apply to the HIBOOS scheme to achieve an EF-HIBOOS scheme [9]. They declare that the schemes they proposed are efficient enough to be practically deployed. However, the authors have not implemented and experimental evaluated the schemes.

In this letter, we consider several HIBS schemes, which have been formally proved secure and have not yet been implemented. The selected schemes include a primitive HIBS scheme proposed by Chow et al. [5], and an escrow-free extension, an online/offline extension, a comprehensive scheme proposed by Chen et al. [4], [8], [9]. We implement the schemes, and evaluate the scheme performance both theoretically and experimentally. We analyze the evaluation results to study the signing/verification delays and transmission overhead of the schemes, as well as the performance influence of the *EF* and *OO* solution.

2. Scheme Review

In this section, we review the construction of four proved secure HIBS schemes, which we implement in this letter and evaluate in Section 3. The schemes are primitive scheme SHER-IBS, escrow-free scheme CWS-EF-HIBS, online/offline scheme CWS-HIBOOS, and comprehensive scheme CWS-EF-HIBOOS.

2.1 SHER-IBS

Let K be the security parameter given to the setup algorithm, and let $\mathcal{G}$ be a BDH parameter generator.

Setup. The root PKG works as follows:

1. runs $\mathcal{G}$ on input K to generate multiplicative groups G_1, G_2 of same prime order, and a bilinear pairing $\hat{e} : G_1 \times G_1 \rightarrow G_2$;
2. chooses random $\alpha \in \mathbb{Z}_p^*$ and two generators $g, g_2 \in G_1$, computes $g_1 = g^\alpha$;
3. randomly picks $h_1, \ldots, h_\ell \in G_1$;
4. chooses cryptographic hash functions $H : \{0,1\}^* \times G_1 \rightarrow \mathbb{Z}_p^*$;

Manuscript received January 1, 2011.

Manuscript revised January 1, 2011.

[†] The author is with the College of Computer, National University of Defense Technology, Changsha, 410073, China

[*] The author is with the National Key Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, 410073, China

a) E-mail: chenpeixin@nudt.edu.cn

DOI: 10.1587/trans.E0.??.1

5. publishes $Param = \{\hat{e}, g, g_1, g_2, h_1, \ldots, h_\ell, H\}$ as public parameters and keeps MSK $= g_2^\alpha$ as master secret.

KeyGen. For an input ID $= \{I_1, \ldots, I_k\}$, the $level_{k-1}$ domain PKG with private key $d_{\mathrm{ID}_{|k-1}} = \{d'_0, \ldots, d'_{k-1}\}$ generates the key d_{ID} as follows:

1. picks random $r_k \in \mathbb{Z}_p^*$;
2. set $d_{\mathrm{ID}} = \{d'_0 F_k(I_k)^{r_k}, d'_1, \ldots, d'_{k-1}, g^{r_k}\}$, where $F_k(x) = g_1^x h_k$.

Signing. To sign a message m with respect to identity ID $= \{I_1, \ldots, I_k\}$, user takes the private key $d_{\mathrm{ID}} = \{d_0, d_1, \ldots, d_k\}$ as input, running the algorithm as follows:

1. picks a random $s \in \mathbb{Z}_p^*$ and computes $x = g_2^s$;
2. computes $h = H(m, x)$;
3. for $j = 1, \ldots, k$, computes $y_j = d_j^{s+h}$;
4. computes $z = d_0^{s+h}$;
5. sets signature as $\sigma = \{x, y_1, \ldots, y_k, z\}$.

Verification. To verify a signature $\sigma = \{x, y_1, \ldots, y_k, z\}$ on message m with respect to identity ID $= \{I_1, \ldots, I_k\}$, the receiver works as follows:

1. computes $h = H(m, x)$;
2. checks $\hat{e}(g, z) \overset{?}{=} \hat{e}(g_1, g_2^h x \prod_{j=1}^k y_j^{I_j}) \prod_{j=1}^k \hat{e}(y_j, h_j)$. If so, outputs 1; otherwise, outputs 0.

2.2 CWS-EF-HIBS

Setup. The root PKG works as follows:

1. generates groups G_1, G_2 and bilinear pairing $\hat{e} : G_1 \times G_1 \rightarrow G_2$;
2. chooses random $\alpha \in \mathbb{Z}_p^*$ and two generators $g, g_2 \in G_1$, computes $g_1 = g^\alpha$;
3. randomly picks $h_1, \ldots, h_\ell \in G_1$;
4. chooses cryptographic hash functions $H_1 : \{0,1\}^* \times G_1 \rightarrow \mathbb{Z}_p^*$ and $H_2 : G_1 \times \{0,1\}^* \rightarrow G_1$;
5. publishes $Param = \{\hat{e}, g, g_1, g_2, h_1, \ldots, h_\ell, H_1, H_2\}$ as public parameters and keeps MSK $= g_2^\alpha$ as master secret.

KeyGen. The same as the *KeyGen* algorithm in SHER-IBS.
Publish. In this phase, user publishes a public parameter and gets PKG signing factor from the root PKG as follows:

1. randomly picks $s_{\mathrm{ID}} \in \mathbb{Z}_p^*$ as user secret and computes $g_{\mathrm{ID}} = g^{s_{\mathrm{ID}}}$;
2. publishes g_{ID} by submitting it to the AP;
3. sends g_{ID} to the root PKG, and gets $f^\alpha = H_2(g_{\mathrm{ID}}, \mathrm{ID})^\alpha$ computed by PKG.

Signing. User takes the private key $d_{\mathrm{ID}} = \{d_0, d_1, \ldots, d_k\}$, secret s_{ID} and PKG signing factor f^α as input, running the algorithm as follows:

1. picks a random $s \in \mathbb{Z}_p^*$ and computes $x = g_2^s$;
2. computes $h = H_1(m, x)$;
3. for $j = 1, \ldots, k$, computes $y_j = d_j^{s+h}$;
4. computes $f = H_2(g_{\mathrm{ID}}, \mathrm{ID})$;
5. computes $z = d_0^{s+h} f^{s_{\mathrm{ID}}} f^\alpha = d_0^{s+h} f^{s_{\mathrm{ID}}+\alpha}$;
6. sets signature as $\sigma = \{x, y_1, \ldots, y_k, z, g_{\mathrm{ID}}\}$.

Verification. The receiver verifies the signature as follows:

1. computes $h = H_1(m, x)$ and $f = H_2(g_{\mathrm{ID}}, \mathrm{ID})$;
2. checks whether $\hat{e}(g, z) = \hat{e}(g_1, g_2^h x \prod_{j=1}^k y_j^{I_j}) \hat{e}(f, g_{\mathrm{ID}} g_1) \prod_{j=1}^k \hat{e}(y_j, h_j)$ holds. If so, outputs 1; otherwise, outputs 0.

Blame. Given $\{\mathrm{ID}, m, \sigma\}$, where $\sigma = \{x, y_1, \ldots, y_k, z, g_{\mathrm{ID}}\}$, this algorithm requires the user parameter g'_{ID} with respect to the identity ID from the AP. It outputs 1 if and only if $g_{\mathrm{ID}} \neq g'_{\mathrm{ID}}$ and 0 otherwise.

2.3 CWS-HIBOOS

Setup. The same as the *Setup* algorithm in SHER-IBS.
KeyGen. The same as the *KeyGen* algorithm in SHER-IBS.
Offline Signing. The signer runs the offline signing algorithm as follows:

1. picks a random $s \in \mathbb{Z}_p^*$ and computes $x = g_2^s$;
2. for $j = 1, \ldots, k$, computes $y_j = d_j^s$;
3. computes $E_{off} = \hat{e}(g_1, \prod_{j=1}^k d_j^{I_j}) \prod_{j=1}^k \hat{e}(d_j, h_j)$;
4. computes $z_{off} = d_0^s$;
5. sets signature as $\sigma_{off} = \{x, y_1, \ldots, y_k, z_{off}, E_{off}\}$.

Online Signing. The signer computes the followings:

1. computes $h = H(m, x)$;
2. computes $z = d_0^h z_{off} = d_0^{s+h}$;
3. computes $E = E_{off}^h$;
4. sets signature as $\sigma_{on} = \{z, E\}$.

The signature is $\sigma = \{x, y_1, \ldots, y_k, z, E\}$.
Verification. The receiver verifies the signature as follows:

1. computes $h = H(m, x)$;
2. checks whether the equation holds:
   $\hat{e}(g, z) = E \cdot \hat{e}(g_1, g_2^h x \prod_{j=1}^k y_j^{I_j}) \prod_{j=1}^k \hat{e}(y_j, h_j)$.
   If so, outputs 1; otherwise, outputs 0.

2.4 CWS-EF-HIBOOS

Setup. The same as the *Setup* algorithm in CWS-EF-HIBS.
KeyGen. The same as the *KeyGen* algorithm in SHER-IBS.
Publish. User publishes a public parameter and gets PKG signing factor from the root PKG as follows:

1. randomly picks $s_{\mathrm{ID}} \in \mathbb{Z}_p^*$ as user secret;
2. computes $g_{\mathrm{ID}} = g^{s_{\mathrm{ID}}}$;
3. publishes g_{ID} by submitting it to the AP;
4. sends g_{ID} to the root PKG, the root PKG computes and returns $f^\alpha = H_2(g_{\mathrm{ID}}, \mathrm{ID})^\alpha$.

Offline Signing. The signer runs the offline signing algorithm as follows:

1. picks a random $s \in \mathbb{Z}_p^*$ and computes $x = g_2^s$;
2. for $j = 1, \ldots, k$, computes $y_j = d_j^s$;
3. computes $f = H_2(g_{\mathrm{ID}}, \mathrm{ID})$;

Table 1: Algorithm computational cost. SHER stands for SHER-IBS scheme, EF for CWS-EF-HIBS scheme, OO for CWS-HIBOOS scheme, EFOO for CWS-EF-HIBOOS scheme.

	Off. Sign	On. Sign	Verify
SHER	–	$(k+2)T_e$	$(k+1)T_e + (k+2)T_p$
EF	–	$(k+3)T_e$	$(k+1)T_e + (k+3)T_p$
OO	$(2k+2)T_e + 2T_p$	$2T_e$	$(k+1)T_e + (k+2)T_p$
EFOO	$(2k+3)T_e + (k+1)T_p$	$2T_e$	$(k+1)T_e + (k+3)T_p$

4. computes $E_{off} = \hat{e}(g_1, \prod_{j=1}^{k} d_j^{I_j}) \prod_{j=1}^{k} \hat{e}(d_j, h_j)$;
5. computes $z_{off} = d_0^s f^{s_{\mathrm{ID}}} f^{\alpha} = d_0^s f^{s_{\mathrm{ID}}+\alpha}$;
6. sets the offline signature as $\sigma_{off} = \{x, y_1, \ldots, y_k, z_{off}, E_{off}, g_{\mathrm{ID}}\}$.

Online Signing. The signer computes the followings:

1. computes $h = H_1(m, x)$;
2. computes $z = d_0^h z_{off} = d_0^{s+h} f^{(s_{\mathrm{ID}}+\alpha)}$;
3. computes $E = E_{off}^h$;
4. sets signature as $\sigma_{on} = \{z, E\}$.

The signature is $\sigma = \{x, y_1, \ldots, y_k, z, E, g_{\mathrm{ID}}\}$.

Verification. The receiver verifies the signature as follows:

1. computes $h = H_1(m, x)$ and $f = H_2(g_{\mathrm{ID}}, \mathrm{ID})$;
2. checks whether the equation holds: $\hat{e}(g, z) = E \cdot \hat{e}(g_1, g_2^h x \prod_{j=1}^{k} y_j^{I_j}) \hat{e}(f, g_{\mathrm{ID}} g_1) \prod_{j=1}^{k} \hat{e}(y_j, h_j)$. If so, outputs 1. Otherwise, outputs 0.

Blame. Given $\{\mathrm{ID}, m, \sigma = \{x, y_1, \ldots, y_k, z, g_{\mathrm{ID}}\}\}$, the algorithm requires the user parameter g'_{ID} with respect to the identity ID from the AP. It outputs 1 if and only if $g_{\mathrm{ID}} \neq g'_{\mathrm{ID}}$ and 0 otherwise.

3. Performance Evaluation

In this section, we evaluate the performance of the above described schemes in terms of signing/verification computational cost and transmission overhead both theoretically and experimentally.

3.1 Theoretically Comparison

We make comparison of computational cost and transmission overhead between the HIBS schemes theoretically.

Let k denote the length of identity, T_e the time to perform a group exponentiation computation and T_p the time of a pairing computation. Since the overhead of hash computation and group point multiplicative computation are negligible comparing with T_e and T_p, we only consider these operations and neglect all the other operations.

Table 1 shows the theoretical performance comparison. From the signing equation in Section 2.1, we can observe that the time to sign a signature in the SHER-IBS is $(k+2)T_e$, which grows linearly as the identity length k. Nevertheless, with the online/offline model, the CWS-HIBOOS scheme can obtain a constant-time online signing algorithm. Although the scheme introduces a high computational cost offline signing algorithm, it has little influence in a real-time scenario since the algorithm is implemented offline. Analyzing the signing/verification equation in Section 2.2, we observe that the escrow-free model only brings in a group exponentiation computation and a paring computation to the signing and verification algorithm, respectively. It is acceptable to solve the key escrow problem. Lowest row of Table 1 shows that the comprehensive scheme obtains both the benefits introduced by *OO* and *EF* model.

Table 2: Transmission overhead. k denotes the length of identity, size stands for the number of group elements, increased ratio is the quotient of the increased signature size divided by size of signature in SHER-IBS scheme.

	Signature size		User param. size
SHER	$k+2$	increased ratio	–
EF	$k+3$	$1/(k+2)$	1
OO	$k+3$	$1/(k+2)$	–
EFOO	$k+4$	$2/(k+2)$	1

Table 2 shows the transmission overhead of the schemes. Here, the transmission overhead includes a signature appended to an original message and a user parameter for key abusing detecting in the *EF* model. We can observe that both the *EF* and *OO* models introduce trivial overhead to the primitive scheme, especially when the k, i.e. the length of identity, is large.

3.2 Experimental Evaluation

The evaluation results were obtained through implementations of the above four schemes based on the PBC library [10]. Our implementations used the curves and parameters defined in Section 2 and were written in C and compiled with GCC 4.8.4 in Ubuntu 14.04. We ran the programs at a Desktop, which equipped Intel(R) Core Duo CPU i5-3550 at 3.30 GHz and 8G memory.

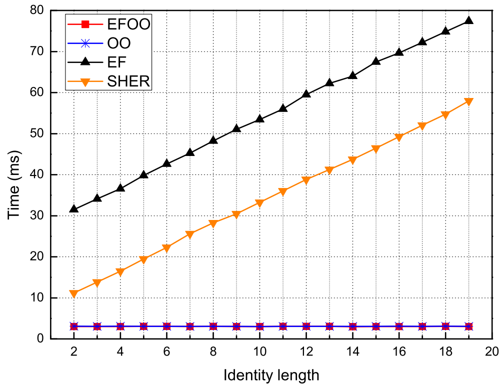

Fig. 1: Computational cost of online signing. Curve “OO” and curve “EFOO” are almost overlapped.

Varying the length of user identity, Figure 1 plots the computational cost of online signing of the HIBS schemes. Note that, we take the signing procedures in SHER-IBS and

CWS-EF-HIBS scheme as online signing since there are no online/offline signing procedures in the schemes. From Figure 1, we can observe that the computational cost of online signing in CWS-HIBOOS and CWS-EF-HIBOOS scheme are constant, which stay around 3 ms. On the other hand, the schemes without *OO* model obtain a linear growth cost of signing. And the lowest cost rises to 30 ms when *EF* model is applied. Thus, we find that applying the *OO* model is an effective way to improve the real-time signing efficiency, especially when the *EF* model has been applied.

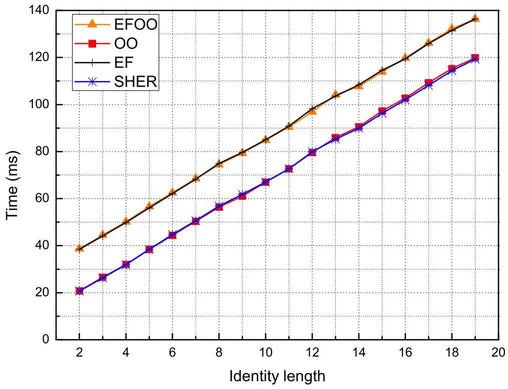

Fig. 2: Computational cost of verification. Curve “EF” and curve “EFOO” are almost overlapped, and so as the curve “SHER” and curve “OO”.

Figure 2 shows the computational cost of verification of the HIBS schemes. From the figure, we can observe that the cost of verification also grows linearly as the length of identity in each scheme, and the OO model does not improve the verification efficiency. Comparing to online signing, the verification computational cost is still quite too high, and applying the EF model will even bring in about 20 ms to the cost. Therefore, we suggest that the verification algorithms need to be further improved.

4. Conclusions

In this work, we reviewed four HIBS schemes including a primitive scheme, an online/offline extension, an escrow-free extension and a comprehensive scheme. We implemented the schemes and evaluated their performance in terms of signing/verification computational cost and transmission overhead both theoretically and experimentally. Evaluation results showed that the escrow-free model proposed by Chen et al. was an effective solution to the key escrow problem. And the online/offline model could improve the real-time signing efficiency.

Acknowledgments

This research was supported by the project of the National Science Foundation of China(NSFC) under grant no. 61303264.

References

[1] A. Shamir, “Identity-based cryptosystems and signature schemes,” Advances in cryptology, pp.47–53, Springer, 1985.

[2] J.C. Choon and J.H. Cheon, “An identity-based signature from gap diffie-hellman groups,” in Public key cryptography – PKC 2003, pp.18–30, Springer, 2002.

[3] C. Gentry and A. Silverberg, “Hierarchical id-based cryptography,” in Advances in cryptology – ASIACRYPT 2002, pp.548–566, Springer, 2002.

[4] P. Chen, X. Wang, and J. Su, “An escrow-free hierarchical identity-based signature model for cloud storage,” Algorithms and Architectures for Parallel Processing - ICA3PP International Workshops and Symposiums, Zhangjiajie, China, November 18-20, 2015, Proceedings, pp.633–647, 2015.

[5] S.S. Chow, L.C. Hui, S.M. Yiu, and K. Chow, “Secure hierarchical identity based signature and its application,” in Information and Communications Security, pp.480–494, Springer, 2004.

[6] S. Even, O. Goldreich, and S. Micali, “On-line/off-line digital signatures,” Advances in Cryptology – CRYPTO89 Proceedings, pp.263–275, Springer, 1990.

[7] Y. Ming and Y. Wang, “Improved identity based online/offline signature scheme,” Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2010 7th International Conference on, pp.126–131, IEEE, 2010.

[8] P. Chen, X. Wang, and J. Su, “An online/offline hibs scheme for privacy protection of people-centric sensing,” Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2015 International Conference on, pp.128–131, Dec 2015.

[9] P. Chen, X. Wang, and J. Su, “An escrow-free online/offline hibs scheme for privacy protection of people-centric sensing [submit],” Security and Communication Networks, 2015.

[10] B. Lynn, H. Shacham, M. Steiner, and J. Cooley, “The pairing-based cryptography library.” https://crypto.stanford.edu/pbc/.