On Two Kinds of Flaws in Some Server-aided Verification Schemes

Zhengjun Cao[1], Lihua Liu[2], and Olivier Markowitch[3]
(Corresponding author: Zhengjun Cao)

Department of Mathematics, Shanghai University[1]
No.99, Shangda Road, Shanghai, China
Department of Mathematics, Shanghai Maritime University[2]
No.1550, Haigang Ave, Pudong New District, Shanghai, China
Computer Sciences Department, Université Libre de Bruxelles[3]
Boulevard du Triomphe CP 212, 1050 Bruxelles, Belgique
(Email: caozhj@shu.edu.cn)
(Received Sept. 30, 2015; revised and accepted Dec. 7, 2015)

Abstract

At Asiacrypt'05, Girault and Lefranc introduced the primitive of server-aided verification (SAV). In the proposed model, the server is assumed to be untrusted but is supposed to not collude with the legitimate prover. At ProvSec'08, Wu et al. have generalized the Girault-Lefranc SAV model by allowing the server to collude with the legitimate prover, and presented two corresponding SAV signature schemes, SAV-BLS-1 and SAV-BLS-2. In this paper, we argue that the SAV-BLS-1 scheme is somewhat artificial because the computational gain in the scheme is at the expense of additional communication costs. This is a common flaw in most outsourcing computation proposals which have neglected the comparisons between the computational gain and the incurred communication costs. We show also that the SAV-BLS-2 scheme is insecure against collusion attacks. It is another common flaw to have the verifier delegate most computations to the server in a way that prevent the verifier to confirm that the returned values are really bound to the signer's public key.

Keywords: Collusion attack, moderate adversary, outsourcing computation, server-aided verification

1 Introduction

Digital signatures and authentication schemes are broadly used in modern chips. The problem of speeding up the prover's or the signer's computations has interested many researchers. At Asiacrypt'05, Girault and Lefranc [7] formally introduced the primitive of server-aided verification (SAV) in order to speed up the verification task of a signature scheme or an identification scheme. They assumed that the verifier has only small computation capabilities while having access to a more powerful, but untrusted server or, equivalently, to a trusted server via a non authenticated communication link.

In a server-aided verification scheme, two kinds of deviating provers should be considered: The cheater who does not know the private key and the legitimate prover who misbehaves in order to make possible some kind of repudiation. For example, in a SAV signature scheme, an illegitimate prover and the server may collaborate in order to let the verifier accept a fake signature. Therefore, a SAV scheme must resist to collusion attacks launched by the server and an illegitimate prover (who may be represented by the same entity).

At Eurocrypt'95, Lim and Lee [13] put forth a generic method based on the "randomization" of the verification equation. However, the equation is only known to the verifier. In 2002, Girault and Quisquater [8] suggested a new approach based on the hardness of factorization and the composite discrete logarithm problem. At TCC'05, Hohenberger and Lysyanskaya [9] considered that an auxiliary server is made of two untrusted softwares which are assumed not to communicate with each other. In 2006, Dijk et al. [6] presented some protocols to speed up fixed-base variable-exponent exponentiation and variable-base fixed-exponent exponentiation using an untrusted computational resource.

At Asiacrypt'05, Girault and Lefranc [7] formally introduced the primitive of server-aided verification. In the model, the server is assumed to be untrusted but without colluding with the legitimate prover. At ProvSec'08, Wu et al. [19] have generalized the Girault-Lefranc SAV model by allowing the server to collude with the legitimate prover, and presented two corresponding SAV signature schemes, SAV-BLS-1 and SAV-BLS-2. They claimed that the SAV-BLS-1 is existentially unforgeable against adap-

tive chosen message attacks, and that the SAV-BLS-2 is sound against collusion attacks launched by the signer and the server. In 2011, Wu et al. [20] have proposed two another SAV signature schemes, SAV-Waters-1 and SAV-Waters-2, based on Waters' scheme [17]. In 2013, Wu et al. [18] have proved that both signature schemes [20] are insecure against collusion attacks launched by the legitimate signer and the server.

Recently, Lee et al. [11] have investigated the problem of cloud server-aided computation for ElGamal elliptic curve cryptosystem. Liao and Hsiao [12] studied the problem of multi-servers aided verification using self-certified public keys for mobile clients. Liu et al. [15] have investigated the problem of identity-based server-aided decryption. Zhang and Sun [22] proposed an ID-based server-aided verification of short signature scheme without key escrow.

In 2013, Canard et al. [3] considered the method for generically transforming a given well-known secure instance of a cryptographic primitive into a secure server-aided version where the server may be corrupted by the adversary. Chow et al. [5] revisited the definition of the security of server-aided verification. In 2014, Canard, Devigne and Sanders [2] provided some efficient ways to delegate the computation of a pairing $e(A, B)$, depending on the status of A and B. Their protocols enable the limited device to verify the value received from the third party by computing one exponentiation. In 2015, Liu et al. [16] considered the problem of server-aided anonymous attribute-based authentication in cloud computing. Very recently, Xiang and Tang [21] have proposed some efficient outsourcing schemes for modular exponentiations with checkability against untrusted cloud servers. Hsien et al. [10, 14] presented two surveys of public auditing for secure data storage in cloud computing. The computation of bilinear pairing represents most of the computing cost when dealing with pairing-based cryptographic protocols. In recent, Chen et al. [4] have put forth a new outsourcing algorithm for bilinear pairings in two untrusted programs model.

In this paper, we argue that the SAV-BLS-1 scheme is artificial because the delegated computations do not represent the heavier computation part; the verification procedure should therefore be rather performed solely by the verifier himself. What appears to be a computational gain in the scheme is due to communication costs that are not taken into account in the analysis. Nevertheless these costs could be far more important than the claimed computational gain. Then we show also that the SAV-BLS-2 scheme is insecure against collusion attacks launched by the server and illegitimate provers. This attack is possible because the verifier delegates its computations to the server in a way that prevent the verifier to confirm that the returned values from an adversary are really bound to the signer's public key. More generally, we point out that most outsourcing computation proposals have neglected the comparisons between the computational gains and the incurred communication costs.

2 The Reasonable Assumptions on SAV Model

Girault and Lefranc [7] assumed that the verifier has only a small computation capability but has access to a more powerful while untrusted server or, equivalently, to a trusted server via an unauthenticated communication link. From the practical of point of view, it appears to be more reasonable to assume that the verifier has access to a trusted server via an unauthenticated communication link, rather than to a potentially malicious server via an authenticated communication link (see Figure 1). This assumption holds on the following considerations:

1) The server is usually set up by some public service agencies. It is trusted due to the credibility of these agencies.

2) The server is assumed to serve many users. It is unrealistic to construct so many authenticated communication links between the server and so many users (chips with small computation capability).

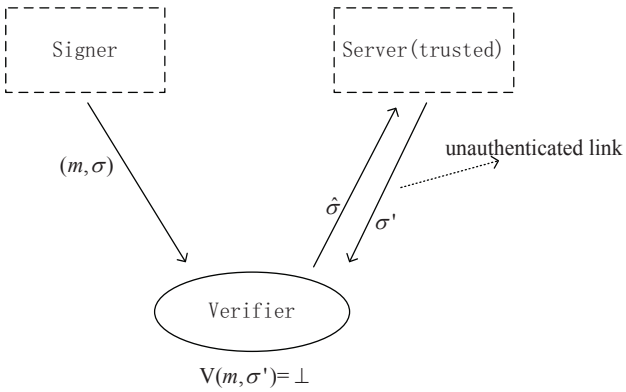

Figure 1: The trusted server with unauthenticated link in SAV model

3 Cryptanalysis of SAV-BLS-1 Signature Scheme

3.1 Review of SAV-BLS-1 Scheme

At ProvSec'08, Wu et al. generalized the Girault-Lefranc SAV model by allowing the server to collude with the legitimate prover, and presented two SAV signature schemes which are based on the BLS scheme [1]: SAV-BLS-1 and SAV-BLS-2. They claimed that the SAV-BLS-1 is existentially unforgeable against adaptive chosen message attacks, and that SAV-BLS-2 is sound against collusion attacks launched by the signer and the server.

The SAV-BLS-1 scheme can be described as follows.

1) ParamGen: Let $(\mathbb{G}_1, \mathbb{G}_T)$ be bilinear groups where $|\mathbb{G}_1| = |\mathbb{G}_T| = p$, for some prime number $p \geq 2^k$, k be the system security number and g be the generator of $\mathbb{G}_1$. e denotes the bilinear map $\mathbb{G}_1 \times$

Table 1: SA-Verify in SAV-BLS-1 signature scheme

Signer		Verifier		Server
$(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$,		Precomputation:		
$pk : y = g^x$; $sk : x$.		$r \in \mathbb{Z}_p$, $R = g^r$		
Given m, compute				
$\sigma = H(m)^x$	$\overset{m,\sigma,y}{--\rightarrow}$	Don't check $e(\sigma, g) = e(H(m), y)$	$\overset{\sigma,R}{--\rightarrow}$	
		Compute $K_2 = e(H(m), y)^r$	$\overset{K_1}{\leftarrow --}$	$K_1 = e(\sigma, R)$
		Check $K_1 \overset{?}{=} K_2$		

$\mathbb{G}_1 \rightarrow \mathbb{G}_T$. There is one cryptographic hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$. The system parameter $param = (\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$.

2) KeyGen: The signer picks a random number $x \in \mathbb{Z}_p^*$ and keeps it as the secret key. The public key is set as $y = g^x$.

3) Sign: For a message m, the signer uses its secret key to generate the signature $\sigma = H(m)^x$.

4) Verify: For a message/signature pair (m, σ), one can check whether $e(\sigma, g) \overset{?}{=} e(H(m), y)$.

5) SA-Verifier-Setup: Given the system parameter $(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$, the verifier V randomly chooses $r \in \mathbb{Z}_p$ and sets $R = g^r$. The VString is (r, R).

6) SA-Verify: The verifier V and the server S interact with each other using the protocol described in Table 1. Note that R is precomputed and the verifier sends the same R to the server in server-aided verification of different message-signature pairs.

3.2 Analysis of SAV-BLS-1 Scheme

The authors claim that with the server's aid, if the parameters are properly selected, the verifier can save about half of the computational costs. However, we stress here that:

- Since the verifier has to compute $e(H(m), y)$ by himself, we are sure that the verifier has the capability to compute the delegated computation $e(\sigma, g)$. Therefore, the computational gain is only the cost for one pairing computation. According to actual effect, *the verifier is asking an equal capacity server for assistance, not a powerful server.*

- The verifier has to interact with the trusted server via a non authenticated link, by sending σ, R and receiving K_1. From the practical point of view, the communication costs (including authentication of the exchanged data, the possible underlying encryption/decryption, the time delay during the interaction, etc.) could be far more than the above computational gain (i.e., the cost of one pairing computation).

Based on these observations, we argue that the *direct verification* requires far less costs. Therefore, the SAV-BLS-1 signature scheme is somewhat unrealistic.

4 Cryptanalysis of SAV-BLS-2 Signature Scheme

4.1 Review of SAV-BLS-2 Scheme

The SAV-BLS-2 can be briefly described as follows.

- The phases of ParamGen, KeyGen, Sign and Verify are the same as that of SAV-BLS-1.

- SA-Verifier-Setup: Given the system parameter $(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$, the verifier computes $K_1 = e(g, g)$.

- SA-Verify: The verifier and the server interact with each other using the protocol described in Table 2.

4.2 Analysis of SAV-BLS-2 Scheme

In the SAV-BLS-2 scheme, the verifier delegates most of the computations to the server via the unauthenticated link. It is easy to find that the scheme is insecure against collusion attacks launched by the server (impersonated by an adversary) and an illegitimate prover, if the transferred data are not authenticated. In fact, the computations $\sigma' = \sigma g^r$, $K_2 = K_3 K_1^r$ performed by the verifier *do not invoke the public key* y. That means the verifier loses the ability to confirm that the returned values from the adversary are really bound to the signer's public key.

We describe here an attack launched by the server and an illegitimate prover: Let the illegitimate prover generate a fake signature $(m, H(m)^\theta)$ for some random $\theta \in \mathbb{Z}_p$ and send it to the verifier. Meanwhile, the prover sends θ to

Table 2: SA-Verify in SAV-BLS-2 signature scheme

Signer		Verifier		Server
$(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$,		Precomputation:		
$pk : y = g^x$; $sk : x$.		$K_1 = e(g, g)$		
Given m, compute				
$\sigma = H(m)^x$	$\overset{m,\sigma,y}{\longrightarrow}$	Don't check $e(\sigma, g) = e(H(m), y)$		
		Pick $r \in \mathbb{Z}_p$, compute $\sigma' = \sigma g^r$	$\overset{m,\sigma',y}{\longrightarrow}$	$K_2 = e(\sigma', g)$
		Check $K_2 \overset{?}{=} K_3 K_1^r$	$\overset{K_2,K_3}{\longleftarrow}$	$K_3 = e(H(m), y)$

Table 3: An attack against SAV-BLS-2 signature scheme

Illegitimate prover		Verifier		Server (impersonated)
$(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$,		Precomputation:		
$pk : y$.		$K_1 = e(g, g)$		
Given m, pick $\theta \in \mathbb{Z}_p$.	-----	----------	$\overset{\theta}{\longrightarrow}$	
Compute $\sigma = H(m)^\theta$	$\overset{m,\sigma,y}{\longrightarrow}$	Pick $r \in \mathbb{Z}_p$,	$\overset{m,\sigma',y}{\longrightarrow}$	$K_2 = e(\sigma', g)$
		compute $\sigma' = \sigma g^r$		
		Check $K_2 \overset{?}{=} K_3 K_1^r$	$\overset{K_2,K_3}{\longleftarrow}$	$K_3 = e(H(m), g^\theta)$

the server. See the following Table 3 for the details of the attack.

Correctness. It is easy to find that

$$\begin{aligned} K_3 K_1^r &= e(H(m), g^\theta) e(g, g)^r \\ &= e(H(m)^\theta, g) e(g^r, g) \\ &= e(H(m)^\theta g^r, g) \\ &= e(\sigma', g) = K_2 \end{aligned}$$

which means the verifier will accept the fake signature.

Remark 1. *If the verifying equation for a signature generated by a user is not bound to its public key, the verifier cannot be convinced that the signature is truly generated by the signer.*

Remark 2. *The adversary (who impersonates the server) himself can play the role of the illegitimate prover (see the modified definition of SAV model in Reference [19]). That means the SAV signature scheme is universally forgeable.*

5 Cryptanalysis of SAV-BLS-3 Signature Scheme

5.1 Review of SAV-BLS-3 Scheme

In 2011, Wu et al. [20] have proposed a new variation of SAV-BLS scheme. We briefly describe it as follows (see Table 4).

5.2 Analysis of SAV-BLS-3 Scheme

The scheme specifies that the computation of $e(H(m), y)$ is performed by the verifier himself. Thus the verification equation $K_2 = K_3^{r_1} K_1^{r_2}$ is really bounded to the signer's public key y and the challengers r_1, r_2 chosen by the verifier. In such case, our attack against the SAV-BLS-2 scheme fails. But we would like to stress that the SAV-BLS-3 scheme has the same flaw as the SAV-BLS-1 scheme. Namely, the verifier is asking an equal capacity server for assistance, not a powerful server.

It is easy to see that the SAV-BLS-3 scheme trades off the cost of one pairing computation for communication costs as well as the costs of some additional exponentiations. It is more inefficient than the SAV-BLS-1 scheme.

6 Further Siscussions

6.1 A Moderate Adversary

In 2013, Chow, Au and Susilo [5] modified the definition of the SAV security in [19, 20]. They clarified the goal of adversaries in the SAV model, and stressed that the adversary may benefit not only when an invalid signature is falsely-claimed as a valid one but also benefit from claiming a valid signature as invalid.

We now want to remark that a moderate adversary may try to make a fake signature but a wicked adversary

Table 4: SA-Verify in SAV-BLS-3 signature scheme

Signer		Verifier		Server
$(\mathbb{G}_1, \mathbb{G}_T, k, g, p, e, H)$,		Precomputation:		
$pk : y = g^x$; $sk : x$.		$K_1 = e(g, g)$		
Given m, compute				
$\sigma = H(m)^x$	$\overset{m,\sigma,y}{--\rightarrow}$	Don't check $e(\sigma, g) = e(H(m), y)$		
		Pick $r_1, r_2 \in \mathbb{Z}_p$, compute $\sigma' = \sigma^{r_1} g^{r_2}$	$\overset{\sigma'}{--\rightarrow}$	
			$\overset{K_2}{\leftarrow --}$	$K_2 = e(\sigma', g)$
		Compute $K_3 = e(H(m), y)$		
		Check $K_2 \overset{?}{=} K_3^{r_1} K_1^{r_2}$		

can ruin all valid signatures. A malicious server can always output some random values such that the verifier fails to check a valid signature. Of course, the latter is out the scope of academic study. From the practical point of view, it is reasonable to assume that the adversary is *moderate* at least: he may cheat the verifier in order to accept an invalid signature, instead of cheating the verifier to reject a valid signature. Otherwise, introducing a fully malicious adversary into the process of verification is meaningless.

Under this assumption, we think the most serious problem related to server-aided verification or outsourcing computation is when the computational gains are less than the incurred communication costs. We have observed that most outsourcing computation proposals had neglected the comparisons between the computational gains and the incurred costs. We argue that it is unnecessary for some SAV schemes to outsource one or two pairing computations at the expense of more communication costs if the verifier himself has the capability to compute the pairings.

6.2 A Nearby and Trusted Server

Girault and Lefranc [7] have described some situations in which a chip with a small computation capability is connected to a powerful device.

- In a GSM mobile telephone, the more sensitive cryptographic operations are performed in the so-called SIM (Subscriber Identification Module), which is already aided by the handset chip, mainly to decipher the over-the-air enciphered conversation.
- In a payment transaction, a so-called SAM (Secure Access Module) is embedded in a terminal already containing a more powerful chip.
- A smart card is plugged into a personal computer, seeing that many PCs will be equipped with smart card readers in a near future.

But we find that in all these situations (a SIM vs. a handset, a SAM vs. a powerful terminal, a smart card vs. a personal computer) the servers are nearby and trusted, not remote and untrusted.

In practice, we think, it is better to consider the scenario where a portable chip has access to a nearby and trusted server, but not to a remote server. Otherwise, the communication costs could overtake the computational gain of the outsourced computations.

7 Conclusion

In this paper we argue that the assumption on malicious server in the SAV model should be interpreted as a trusted server with unauthenticated channels, not an untrusted server with authenticated channels. We then argue that it incurs more communication costs which cannot be simply neglected and make most SAV schemes unpractical. We would like to stress that in a SAV signature scheme one has to balance carefully the delegated computations and the verifier's computations.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Project 61303200, 61411146001) and the project of CRYPTASC (funded by the Brussels Institute for Research and Innovation). The authors gratefully acknowledge the reviewers for their valuable suggestions.

References

[1] D. Boneh, G. Lynn, and H. Shacham, "Short signature from the weil pairing," in *Proceedings of Advances in Cryptology (Asiacrypt'01)*, pp. 514–532, Gold Coast, Australia, December 2001.

[2] S. Canard, J. Devigne, and O. Sanders, "Delegating a pairing can be both secure and efficient," in *Proceedings of Applied Cryptography and Network Security*, pp. 549–565, Lausanne, Switzerland, June 2014.

[3] S. Canard and et al., "Toward generic method for server-aided cryptography," in *Proceedings of Information and Communications Security*, pp. 373–392, Beijing, China, November 2013.

[4] X. F. Chen and et al., "Efficient algorithms for secure outsourcing of bilinear pairings," *Theoretical Computer Science*, no. 562, pp. 112–121, 2015.

[5] S. Chow, M. H. Au, and W. Susilo, "Server-aided signatures verification secure against collusion attack," *Information Security Technical Report*, vol. 17, pp. 46–57, 2013.

[6] M. Dijk and et al., "Speeding up exponentiation using an untrusted computational resource," *Designs, Codes and Cryptography*, vol. 39, pp. 253–273, 2006.

[7] M. Girault and D. Lefranc, "Server-aided verification: Theory and practice," in *Proceedings of Advances in Cryptology (Asiacrypt'05)*, pp. 605–623, Chennai, India, December 2005.

[8] M. Girault and J. Quisquater, "Gq + gps = new ideas + new protocols," in *Proceedings of Advances in Cryptology (Eurocrypt'02)*, Amsterdam, Netherlands, May 2002.

[9] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in *Proceedings of Theory of Cryptography*, pp. 264–282, Cambridge, MA, USA, February 2005.

[10] W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[11] N. Y. Lee, Z. L. Chen, and F. K. Chen, "Cloud server aided computation for elgamal elliptic curve cryptosystem," in *Proceedings of IEEE 37th Annual Workshops of Computer Software and Applications Conference*, pp. 11–15, Kyoto, Japan, July 2013.

[12] Y. P. Liao and C. M. Hsiao, "A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients," *Future Generation Computer Systems*, vol. 29, pp. 886–900, 2013.

[13] C. H. Lim and P. J. Lee, "Server (prover/signer)-aided verification of identity roofs and signatures," in *Proceedings of Advances in Cryptology (Eurocrypt'95)*, pp. 64–78, Saint-Malo, France, May 1995.

[14] C. W. Liu, W. F. Hsien, C. C. Yang, and M. S. Hwang, "A survey of public auditing for shared data storage with user revocation in cloud computing," *International Journal of Network Security*, vol. 18, no. 4, pp. 650–666, 2016.

[15] J. Liu, C. K. Chu, and J. Y. Zhou, "Identity-based server-aided decryption," in *Proceedings of Information Security and Privacy (Acisp'11)*, pp. 337–352, Melbourne, Australia, July 2011.

[16] Z. S. Liu, H. Y. Yan, and Z. K. Li, "Server-aided anonymous attribute-based authentication in cloud computing," *Future Generation Computer Systems*, vol. 52, pp. 61–66, 2015.

[17] B. Waters, "Efficient identity-based encryption without random oracles," in *Proceedings of Advances in Cryptology (Eurocrypt'05)*, pp. 114–127, Aarhus, Denmark, May 2005.

[18] H. Wu, C. X. Xu, J. Deng, and J. B. Ni, "On the security of two server-aided verification signature schemes," *Journal of Computational Information Systems*, vol. 9, no. 4, pp. 1449–1454, 2013.

[19] W. Wu, Y. Mu, W. Susilo, and X. Y. Huang, "Server-aided verification signatures: Definitions and new constructions," in *Proceedings of Provable Security*, pp. 141–155, Shanghai, China, November 2008.

[20] W. Wu, Y. Mu, W. Susilo, and X. Y. Huang, "Provably secure server-aided verification signatures," *Computers and Mathematics with Applications*, vol. 61, no. 7, pp. 1705–1723, 2011.

[21] C. Xiang and C. M. Tang, "Efficient outsourcing schemes of modular exponentiations with checkability for untrusted cloud server," *Journal of Ambient Intelligence and Humanized Computing*, no. 6, pp. 131–139, 2015.

[22] J. H. Zhang and Z. B. Sun, "An ID-based server-aided verification short signature scheme avoid key escrow," *Journal of Information Science and Engineering*, vol. 29, pp. 459–473, 2013.

Zhengjun Cao is an associate professor of Department of Mathematics at Shanghai University. He received his Ph.D. degree in applied mathematics from Academy of Mathematics and Systems Science, Chinese Academy of Sciences. He served as a post-doctor in Computer Sciences Department, Université Libre de Bruxelles, from 2008 to 2010. His research interests include cryptography, discrete logarithms and quantum computation.

Lihua Liu is an associate professor of Department of Mathematics at Shanghai Maritime University. She received her Ph.D. degree in applied mathematics from Shanghai Jiao Tong University. Her research interests include combinatorics, cryptography and information security.

Olivier Markowitch is an associate professor of the Computer Sciences Department at the Université Libre de Bruxelles. He is also information security advisor of his University. He is working on the design and analysis of two-party and multi-party cryptographic protocols as well as on the design and analysis of digital signature schemes.