2012 Fourth International Conference on Intelligent Networking and Collaborative Systems

Toward Quantum-resistant Strong Designated Verifier Signature from Isogenies

Xi Sun*, Haibo Tian† and Yumin Wang*
*State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, 710071, China
Email:{sunxi,ymwang}@xidian.edu.cn
†School of Information Science and Technology, Sun Yat-Sen University, Guangzhou, 510275, China
Email: tianhb@mail.sysu.edu.cn

***Abstract*—This paper proposes a strong designated verifier signature (SDVS) based on a recently proposed mathematical problem. It consists in searching for an isogeny between supersingular elliptic curves. The problem is hypothetically strong against a quantum computer. This makes our proposal the first SDVS scheme that may be secure against a quantum computer.**

***Keywords*-SDVS; quantum computer; Isogenies;**

I. Introduction

Jakobsson et al. [5] proposed the concept of designated verifier signature (DVS). A DVS consists of a proof that either "the signer has signed on a message" or "the signer has the verifier's secret key". If a designated verifier is confident that her/his private key is kept in secret, the verifier makes sure that a signer has signed on a message. No other parties can be convinced by the DVS since the designated verifier can generate it with her/his private key. It is useful in various commercial cryptographic applications, such as e-voting, copyright protection.

A strong DVS (SDVS) is an extension of the DVS. In the appendix, Jakobsson et al. [5] gave a definition of SDVS. It means that a verifier needs to use her/his private key to verify the signature. It considers a situation where a signature is captured before reaching a designated verifier. In this case, an adversary can know who is the real signer as there are only two possibilities. Laguillaumie and Vergnaud [7], and Saeednia [9] both formalized the notion.

Most SDVS schemes are based on two general mathematical problems: determination of order and structure of a finite Abelian group, and discrete logarithm computation in a cyclic group with computable order [8]. Both of the problems can be solved in a polynomial time using Shor's algorithm for a quantum computer [1]. Then the development of SDVS, which would be strong against a quantum computer, is necessary.

This paper focuses on SDVS schemes secure against a quantum computer.

A. Contribution

Jao and Feo [6] proposed a key exchange protocol by exploiting isogenies between supersingular elliptic curves. Huang et al. [3] showed a method to construct an SDVS by a Diffie-Hellman key exchange protocol. This paper combines the two ideas to give an SDVS scheme based on isogenies.

The construction shows the possibility to construct a kind of signature scheme based on isogenies. Note that current proposed schemes [6], [8] include encryption schemes and key exchange schemes, and no signature schemes. Although our construction heavily relies on the underlying key exchange protocol, and only a designated verifier can verify a signature, it fulfills a kind of signature scheme.

B. Related Works

Rostovtsev and Stolbunov [8] proposed a public key crypto-system based on isogenies. It discussed theoretical background and a public key encryption technique. Stolbunov [10] constructed public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. Childs et al. [2] showed that the private keys in Stolbunov's system can be recovered in sub-exponential time. Jao and Feo [6] proposed an key exchange protocol and new assumptions about quantum resistance.

There are many SDVS schemes. However, the closely related one is the scheme proposed by Huang et al. in [3]. They proposed a short SDVS scheme based on a gap bilinear Diffie-Hellman problem. A signature is simply a keyed message authentication code and the key is a long-term static key between a signer and a designated verifier.

C. Organizations

The next section gives some preliminaries about assumptions and SDVS. Section 3 is the SDVS scheme. The security analysis is in Section 4. A conclusion is in Section 5.

II. Preliminaries

A. Assumptions

Let $p = l_A^{e_A} l_B^{e_B} f \pm 1$ be a prime, where l_A and l_B are small primes, and f is a cofactor such that p is prime. Let E_0 be a supersingular curve over a field $\mathbb{F}_{p^2}$. Let $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ be bases of $E_0[l_A^{e_A}]$ and $E_0[l_B^{e_B}]$, respectively.

Supersingular Computational Diffie-Hellman (SSCDH) problem [6]: Let $\phi_A : E_0 \to E_A$ be an isogeny whose kernel is equal to $\langle [m_A]P_A + [n_A]Q_A \rangle$ for $m_A, n_A \in_R \mathbb{Z}/l_A^{e_A}\mathbb{Z}$, not both divisible by l_A. Let $\phi_B : E_0 \to E_B$ be an isogeny whose kernel is $\langle [m_B]P_B + [n_B]Q_B \rangle$ for

978-0-7695-4808-1/12 $26.00 © 2012 IEEE
DOI 10.1109/iNCoS.2012.70

IEEE computer society

$m_B, n_B \in_R \mathbb{Z}/l_B^{e_B}\mathbb{Z}$, not both divisible by l_B. Given the curves E_A, E_B and the points $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, $\phi_B(Q_A)$, find the j-invariant of $E_0/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle$.

Supersingular Decision Diffie-Hellman (SSDDH) problem [6]: Given a tuple sampled with probability $1/2$ from one of the following two distributions:

- (E_A, E_B, $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$,$\phi_B(Q_A)$, E_{AB}), where E_A, E_B, $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, $\phi_B(Q_A)$ are as in the SSCDH problem, and E_{AB} is in
$$E_0/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle,$$
- (E_A, E_B, $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, $\phi_B(Q_A)$, E_C), where E_A, E_B, $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, $\phi_B(Q_A)$ are as in the SSCDH problem, and E_C is in
$$E_0/\langle [m'_A]P_A + [n'_A]Q_A, [m'_B]P_B + [n'_B]Q_B \rangle,$$
where m'_A, n'_A, m'_B, and n'_B are chosen randomly,

determine from which distribution the triple is sampled.

The SSCDH (SSDDH) assumption is that there are no polynomial time algorithms to solve the SSCDH (SSDDH) problem with a non-negligible probability ϵ in time t.

B. SDVS

We define an SDVS scheme as follows.

- $Setup$: A probabilistic polynomial time algorithm, on inputting a security parameter $\kappa \in \mathbb{Z}$, produces system parameters sp.
- $KeyGen$: A probabilistic polynomial time algorithm, on inputting the system parameters sp, produces key pairs (y_S, x_S) for a signer S, and (y_V, x_V) for a verifier V.
- $Sign$: A probabilistic polynomial time algorithm, on inputting the system parameters sp, a signer's private key x_S, a verifier's public key y_V and a message m, produces a signature δ.
- Ver: A deterministic polynomial time algorithm, on inputting the system parameters sp, a public key y_S of a signer, a private key x_V of a verifier, a message m, and a signature δ, produces a verification decision.
- Sim: A probabilistic polynomial time algorithm, on inputting the system parameters sp, a public key y_S of a signer, a private key x_V of a verifier, and a message m, produces a signature δ.

Properties

We consider three properties of an SDVS scheme, namely unforgeability, non-transferability, and privacy of signer's identity. The following definitions mainly refers to [4].

- Unforgeability: The unforgeability means that if an adversary can produce a signature related to a signer and a verifier, and it knows no private keys of the signer or verifier, it can be used as a black box to solve a hard problem. As the hard problem is not easy to be solved, the premise is false so the adversary cannot produce a valid signature. The concept is formally defined by a game between an adversary $\mathcal{A}$ and a simulator $\mathcal{S}$:
  - $\mathcal{S}$ provides $\mathcal{A}$ system parameters sp, a public key y_S, and a public key y_V.
  - $\mathcal{A}$ adaptively issues queries to the following oracles for polynomially many times:
    - Σ: Given a message m, it returns a valid signature δ with respect to y_S and y_V.
    - Υ: Give a signature δ on a message m, it returns a decision about its validity with respect to y_S and y_V.
  - $\mathcal{A}$ produces a forgery δ^* for a message m^*. It wins the game if the signature is valid for m^* with respect to y_S and y_V, and it has not queried the message m^* to oracle Σ.

  Definition 1: An SDVS scheme is (t, ϵ)-unforgeable, if no adversary $\mathcal{A}$ wins the game with a probability at least ϵ in time at most t.
- Non-Transferability: The non-transferability means that given a valid message-signature pair (m, δ) for a designated verifier, it is infeasible for any probabilistic polynomial-time distinguisher to tell the message was signed by a signer or the designated verifier. The concept is formally defined as follows:
  Definition 2: An SDVS scheme is non-transferable if signatures produced by a signer are computationally indistinguishable from those produced by a designated verifier, i.e.
$$\{Sign(sp, x_S, y_V, m)\} \approx \{Sim(sp, y_S, x_V, m)\}.$$
  If the distributions of the two sets are identical, it is perfect non-transferable.
- Privacy of Signer's Identity: It considers two signers who produce signatures for a designated verifier. Basically, it requires that given a message-signature pair (m, δ), an distinguisher without the private key of the designated verifier, cannot tell it is produced by which signer. The concept is formally defined by a game between an distinguisher $\mathcal{D}$ and a simulator $\mathcal{C}$:
  - $\mathcal{C}$ provides a system parameter sp, two signers' public keys y_{S0} and y_{S1}, and a verifier's public key y_V.
  - $\mathcal{D}$ adaptively issues queries to the following oracles for polynomially many times:
    - Σ_0 or Σ_1: Given a message m, it returns a valid signature δ with respect to y_{S0} and y_V or to y_{S1} and y_V.
    - Υ: Given a message m, a signature δ, and an identity $Sd \in \{S0, S1\}$, it returns a decision about its validity with respect to y_{Sd} and y_V.
  - $\mathcal{D}$ produces a message m^*. $\mathcal{C}$ then flips a fair coin $b^* \leftarrow \{0, 1\}$, and computes a challenge signature $\delta^* = Sign(sp, x_{Sb^*}, y_V, m^*)$. It returns δ^* to $\mathcal{D}$.

– $\mathcal{D}$ continues to issue queries as before except that it could not query Υ on inputting $(m^*, \delta^*, S0)$ or $(m^*, \delta^*, S1)$. Finally, it produces a bit b' and wins the game if $b' = b^*$.

Definition 3: An SDVS scheme is (t, ϵ) secure about privacy of signer's identity if no distinguisher $\mathcal{D}$ wins the game above with probability that deviates from one-half by more than ϵ in time at most t.

III. The SDVS

- $Setup$: Let $(p = l_A^{e_A} l_B^{e_B} f \pm 1,\ E_0,\ \{P_A, Q_A\},\ \{P_B, Q_B\})$ be defined as before. Let $H : \{0,1\}^* \to \{0,1\}^k$ be a secure hash function, where k is a security parameter.
- $KeyGen$: A signer selects at random $m_S, n_S \in_R \mathbb{Z}/l_A^{e_A}\mathbb{Z}$, not both divisible by l_A. It computes an isogeny $\phi_S : E_0 \to E_S$, and computes $\phi_S(P_B)$ and $\phi_S(Q_B)$. The public key of the signer is $(E_S, \phi_S(P_B), \phi_S(Q_B))$. The private key is (m_S, n_S). Similarly, the public key of a designated verifier is $(E_V, \phi_V(P_A), \phi_V(Q_A))$. The private key is (m_V, n_V).
- $Sign$: To sign a message m for a designated verifier V, a signer S does as follows:
  1) Compute an isogeny $\phi'_S : E_V \to E_{SV}$ having kernel equal to $\langle [m_S]\phi_V(P_A) + [n_S]\phi_V(Q_A) \rangle$.
  2) Compute $\delta = H(m || j(E_{SV}))$, where "$||$" denotes bits concatenation, and $j(\cdot)$ is to compute the j-invariant of a elliptic curve.
- Ver: After receiving a signature δ and a message m from a signer S, a verifier V does as follows:
  1) Compute an isogeny $\phi'_V : E_S \to E_{VS}$ having kernel equal to $\langle [m_V]\phi_S(P_B) + [n_V]\phi_S(Q_B) \rangle$.
  2) Compute $\delta' = H(m || j(E_{SV}))$, where "$||$" denotes bits concatenation, and $j(\cdot)$ is to compute the j-invariant of an elliptic curve.
  3) Check whether $\delta = \delta'$.
- Sim: To simulate a signature on m, the verifier behaves the same as in the Ver algorithm to compute a signature δ for a message m.

IV. Proofs

We below use symbols q_s, q_v, q_h to denote the number of query on a signing, verifying, hashing oracles, respectively.

1) Unforgeability:

Proposition 1: If the SSCDH problem is (ϵ, t)-holding, and the hash function is a random oracle, the above SDVS scheme is (ϵ', t')-holding, where

$$\epsilon \geq \frac{\epsilon'}{3 + (q_h + q_s + q_v)\epsilon'}$$

and

$$t' \approx t.$$

Proof: Suppose an adversary $\mathcal{A}$ who can produce valid SDVS signatures. It takes a system parameter, public keys of a signer and a verifier, and queries a signing oracle Σ and a verifying oracle Υ, and a hashing oracle $\mathcal{H}$. Suppose a simulator $\mathcal{S}$ who provides inputs and oracles for $\mathcal{A}$. The simulator tries to solve an SSCDH problem. It takes an instance $(E_A, \phi_A(P_B), \phi_A(Q_B), E_B, \phi_B(P_A), \phi_B(Q_A))$. And then $\mathcal{S}$ plays with $\mathcal{A}$.

- $\mathcal{S}$ sets a signer's public key as $(E_A, \phi_A(P_B), \phi_A(Q_B))$, and sets a designated verifier's public key as $(E_B, \phi_B(P_A), \phi_B(Q_A))$.
- Simulator $\mathcal{S}$ provides a signing oracle Σ and a verifying oracle Υ and a hashing oracle $\mathcal{H}$ as follows.
  – $\mathcal{H}$: There is a hashing table H_t. It is empty at the beginning. On a query $(m_i||X)$, if m_i is not in the table H_t, it selects at random $\delta_i \in_R \{0,1\}^k$ that is not in the table H_t, records an entry (m_i, X, δ_i) in H_t, and returns δ_i as a reply. If (m_i, X) is in the table, it returns δ_i in the matching entry.
  – Σ: On inputting a message m_i, $\mathcal{S}$ selects a random $\delta_i \in_R \{0,1\}^k$, and records an entry $(m_i, \perp, \delta_i)$ in the hashing table H_t. It returns δ_i as a signature for m_i.
  – Υ: On inputting a signature δ for a message m, $\mathcal{S}$ finds a matching entry in the hashing table where $m_i = m$ and $\delta_i = \delta$ and the middle element is $\perp$. If it finds such an entry, the signature is valid. Else, it is invalid.

If $\mathcal{A}$ produces a forged signature δ^* for a message m^* as its final output, $\mathcal{S}$ tries to find a matching entry (m^*, X^*, δ^*) in the table H_t, and returns the value X^* as an answer to the SSCDH problem instance.

For each query on a signing, hashing, or verifying oracle, if $\mathcal{A}$ has guessed a hashing value correctly, or it has solved the SSCDH problem, or it has guessed a j-invariant correctly, an oracle simulation fails. So the successful simulation happens with a probability at least

$$\begin{array}{rl} & (1 - q_q\epsilon)(1 - \frac{q_q}{2^{|\mathbb{F}_{p^2}|}})(1 - \frac{q_q}{(2^k - q_h - q_s)}) \\ \geq & \frac{4}{9}(1 - q_q\epsilon), \end{array}$$

where $q_q = q_h + q_s + q_v$, and $\frac{q_q}{2^{|\mathbb{F}_{p^2}|}} < 1/3$, and $\frac{q_q}{(2^k - q_h - q_s)} < 1/3$, and $|\mathbb{F}_{p^2}|$ denotes the binary bit-length of an element in $\mathbb{F}_{p^2}$.

When a simulation is successful, $\mathcal{A}$ should produce a signature for a new message m^*. As the message is new, except a random guess, $\mathcal{S}$ can solve the SSCDH problem instance with a probability

$$\begin{array}{rcl} \epsilon & \geq & \epsilon'(1 - \frac{1}{2^k})\frac{4}{9}(1 - q_q\epsilon) \\ & \geq & \frac{1}{3}\epsilon'(1 - q_q\epsilon), \end{array}$$

where $\frac{1}{2^k} < 1/4$.

Then

$$\epsilon \geq \frac{\epsilon'}{3 + (q_h + q_s + q_v)\epsilon'}.$$

The runtime of $\mathcal{S}$ is almost the same as the runtime of $\mathcal{A}$ if we omit the time for table searching and random number generation. ■

2) Non-transferability:

Proposition 2: The SDVS scheme is perfect non-transferable.

This is obvious as a simulated signature is produced in the same way as a real signature if and only if a signer and a designated verifier compute the same shared long-term key, which has been shown correct by Jao and Feo in [6].

3) Privacy of Signer's Identity:

Proposition 3: If the SSDDH problem is (ϵ, t)-holding, a hashing function is modeled as a random oracle, the above SDVS scheme is (ϵ', t') secure, where

$$\epsilon > \epsilon'/8$$

and

$$t \approx t'.$$

Proof: Suppose a distinguisher $\mathcal{D}$ to distinguish a real signer of a given SDVS signature. $\mathcal{D}$ takes two signers' public keys and a verifier's public key as input, and queries two signing oracles Σ_0 and Σ_1, and a verifying oracle Υ, and a hashing oracle $\mathcal{H}$. Suppose a simulator $\mathcal{C}$ provides these oracles and inputs to $\mathcal{D}$. $\mathcal{C}$ tries to solve an SSDDH problem. It takes an instance $(E_A, \phi_A(P_B), \phi_A(Q_B), E_B, \phi_B(P_A), \phi_B(Q_A), E_X)$. Then $\mathcal{C}$ plays with $\mathcal{D}$ as follows.

- Suppose two signers have identities $S0$ and $S1$. $\mathcal{C}$ produces a private key (m_{S0}, n_{S0}) as the private key of $S0$. The public key $(E_{S0}, \phi_{S0}(P_B), \phi_{S0}(Q_B))$ is computed according to the $KeyGen$ algorithm. It sets the public key of $S1$ as $(E_A, \phi_A(P_B), \phi_A(Q_B))$. It also sets a designated verifier's public key as $(E_B, \phi_B(P_A), \phi_B(Q_A))$.
- $\mathcal{C}$ provides $\mathcal{D}$ oracles as follows.
  - $\mathcal{H}$: It is the same as a hashing oracle in the unforgeability proof.
  - Σ_0: It uses $S0$'s private key to produce signatures.
  - Σ_1: It is simulated in the same way as a signing oracle in the unforgeability proof.
  - Υ: If a query is about $S0$ and the designated verifier, $\mathcal{C}$ uses $S0$'s private key to check its validity. Else its validity is checked in the same way as the verifying oracle in the unforgeability proof.
- When $\mathcal{D}$ submits a message m^*, $\mathcal{C}$ randomly flips a coin $d \in \{0, 1\}$.
  - If $d = 0$, it produces a signature by $S0$'s private key for m^*, and returns the signature to $\mathcal{D}$.
  - Else, it produces a signature by computing $\delta = H(m^*, j(E_X))$.
- Then $\mathcal{D}$ continues to query various oracles. The oracle Υ does not response a query on the message-signature pair (m^*, δ^*) with an identity $id \in \{S0, S1\}$.

When $\mathcal{D}$ produces a bit d', if $d' = d$, $\mathcal{C}$ produces an output 1 to indicate that $E_X = E_{AB}$. If $d' \neq d$, $\mathcal{C}$ produces an output 0. If an oracle simulation fails, $\mathcal{D}$ may produce no final output, and $\mathcal{C}$ simply produces an output randomly.

If $E_X = E_{AB}$, $\mathcal{C}$ provides $\mathcal{D}$ valid simulation, $\mathcal{D}$ should show its advantage. If $E_X \neq E_{AB}$, $\mathcal{D}$ obtains a valid signature with a probability about $1/2 + 1/2^k$. If the signature is invalid, the adversary has no advantage. Then the advantage of $\mathcal{C}$ is

$$\begin{aligned}\epsilon &= |Pr[d' = d \wedge E_X = E_{AB}] \\ &\quad - Pr[d' = d \wedge E_X \neq E_{AB}]| \\ &= 1/2|Pr[b' = b^*|E_X = E_{AB}] \\ &\quad - Pr[b' = b^*|E_X \neq E_{AB}]| \\ &= 1/2|(1/2 + \epsilon') - ((1/2 + 1/2^k)(1/2 + \epsilon') \\ &\quad + (1/2 - 1/2^k)/2)| \\ &> \epsilon'(1/4 - 1/2^{k+1}) \\ &> \epsilon'/8,\end{aligned}$$

where $1/2^k < 1/4$.

$\mathcal{C}$ does not need complex computation to simulate oracles. So the runtime of $\mathcal{C}$ is almost equal to the runtime of $\mathcal{D}$. ■

V. Conclusion

We have shown an SDVS scheme that may be secure in the post-quantum era. The construction uses a traditional idea to exploit a key agreement protocol. The proofs show the relations of the assumptions and the properties of an SDVS.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. 61003244), Fundamental Research Funds for the Central Universities (Grant No. 11lgpy71), Doctoral Fund of Ministry of Education of China for New Teachers (Grant No. 20090171120006). We are grateful to the anonymous referees for their invaluable suggestions.

References

[1] D. Boneh, R. Lipton, *Quantum Cryptanalysis of Hidden Linear Functions*, Crypto 1995, LNCS 963, pp. 424–437, 1995.

[2] A. Childs, D. Jao, and V. Soukharev, *Constructing elliptic curve isogenies in quantum subexponential time*, 2010. http://arxiv.org/abs/1012.4019/.

[3] X. Huang, W. Susilo, Y. Mu, and F. Zhang, *Short Designated Verifier Signature Scheme and Its Identity-based Variant*, International Journal of Network Security, Vol.6, No.1, pp. 82–93, 2008.

[4] Q. Huang, G. Yang, D. Wong, and W. Susilo, *Efficient Strong Designated Verifier Signature Schemes without Random Oracles or Delegatability*, Cryptology ePrint Archive: Report 2009/518, 2009.

[5] M. Jakobsson, K. Sako and R. Impagliazzo, *Designated Verifier Proofs and Their Applications*, In Maurer, U.M. (Ed.): EUROCRYPT '96, LNCS 1070, Springer, Saragossa, Spain, May, pp.143–154, 1996.

[6] D. Jao and L. Feo. *Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies Post-Quantum Cryptography*, In: B. Yang (ed.) Post-Quantum Cryptography, LNCS 7071, pp.19–34, Springer Heidelberg, 2011.

[7] F. Laguillaumie, and D. Vergnaud, *Designated verifiers Signature: anonymity and efficient construction from any bilinear map*, In C. Blundo, S. Cimato (Eds.): SCN2004, LNCS 3352, Springer, Amalfi, Italy, September, pp. 105–119, 2005.

[8] A. Rostovtsev, and A. Stolbunov, *Public-key cryptosystem based on isogenies*, 2006. http://eprint. iacr.org/2006/145/.

[9] S. Saeednia, S. Kramer and O. Markovitch, *An efficient strong designated verifier signature scheme*, In J. Lim, D. Lee (Eds.): ICISC 2003, LNCS 2971, Springer, Seoul, Korea, November, pp. 40–54, 2004.

[10] A. Stolbunov, *Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves*, Adv. Math. Commun., 4(2):215–235, 2010.