Introducing Decryption Authority into PKI

Feng Bao
Kent Ridge Digital Labs
21 Heng Mui Keng Terrace
Singapore 119613
baofeng@krdl.org.sg

Abstract

It is well-known that CA plays the central role in PKI. In this paper we introduce a new component into PKI, DA (decryption authority), which decrypts important and sensitive messages for clients under certain conditions. A PKI with DA provides solutions to many security problems in e-commerce and on-line transactions. If we consider that public key cryptography provides both digital signature and asymmetric encryption technologies, DA completes PKI by adding the missing half function. More importantly, DA can greatly increase PKI implementation service revenue. In this paper, we describe the application background and technical principle of DA, give a general explanation on how DA serves clients, and review some relevant research work. We believe that the PKI with DA has great potential to lead to a killing product for e-commerce security.

1 PKI Including CA and DA

Public Key Cryptography

In order to communicate, collaborate and access sensitive information securely in both Internet and Intranet environment, there are a number of security services which need to be provided to protect resources from possible security threats. They include user ID/authentication, confidentiality, access control, integrity, and non-repudiation. Encryption and digital signature offer solutions to implement these security services. Public key cryptography uses a pair of mathematically related keys. One of the key is made available to the public (public key), and the other is kept private (private key). Public key cryptography, which includes both digital signature and asymmetric encryption capabilities, can play an integral role in countering security attacks by providing end-to-end security of information, in terms of confidentiality, integrity, and proof of origin.

PKI with CA

A public key infrastructure (PKI) is normally used to automatically manage public keys through the use of public key digital certificates and digital certificate authorities. A public key digital certificate is a digital document that certifies the association between the identity of an individual, company or server to a public/private key set. To protect from the possible forgery of the certificate, a trusted certificate authority (CA) is needed to vouch for the identities of individuals, companies, or servers to whom it issues certificates. It is necessary that the CA's public key be securely distributed manually, from a trusted directory server, or built in as a trusted part of the program. The PKI addresses requirements that relate to the generation and distribution of keys, the obtaining of public key certificates and the distribution of Certification Revocation Lists (CRLs) which are the hot lists of the canceled certificates. In order to enhance the pure software-solution to the PKI/CA technology, the public key crypto smart cards can be used as tamper-resistant devices to store private keys and digital certificates to improve the flexibility and mobility of the users. There are many companies that provide certificate-based authentication. Many of their systems implement ITU X.509 v.3 standard which is a standard format for public key certificates and X.509 Certificate Revocation List (CRL) v.2 standard CRLs.

PKI with DA

In some practical environments, a trusted authority who conducts decryption is demanded. We call it DA (decryption authority). It is a frequently occurred situation where a party A is willing to give some precious and important messages to another party B under certain conditions. To guarantee the messages reach B only under those conditions, A can encrypts the messages by DA's public key and give the ciphertexts to B, while B can obtain the messages by asking DA to decrypt the ciphertexts for him. DA only does it un-

der those conditions. The conditions may include who should be the recipients of the messages and when the messages should be received, etc. A difficult problem here is that sometimes B should be convinced that the ciphertexts from A are indeed the encryption of those messages instead of some garbage messages. A new technique in public key cryptography can solve this problem.

DA for Digital Signatures

It is widely believed that digital signatures will play a very important role in on-line business in the future. Many countries are making digital signatures legal to support e-commerce security. Digital signatures address the authentication, integrity, and non-repudiation requirements. Considering the above-mentioned messages to be signatures, we need public encryption schemes in which people can prove to others that a ciphertext is an encryption of a required signature. Technically, such public key encryptions schemes exist, which are called verifiable encryption schemes (VES). Therefore, DA's public key should be the public encryption key of VES. With appropriate application of DA, many security problems can be perfectly solved in the on-line business where digital signatures play key roles. We will give some concrete examples of application of DA later in the paper.

Compare CA and DA

Both CA and DA are trusted authorities, but they play different roles. CA conducts signing while DA conducts decrypting. They exploit the two aspects of public key cryptography, namely, digital signature and asymmetric encryption. In the viewpoint of the objects, CA deals with user's public keys while DA deals with user's digital signatures. The output from CA is the so-called certificate, i.e., the signature by CA's private key, while the output from DA is the plaintext decrypted from DA's private key. Both CA and DA must have public keys, which are the root of the trust and are supposed to be correctly known by everyone. From business viewpoint, DA has more prospective profit than CA since DA is more often used.

2 A Brief Description of VES

A verifiable encryption scheme (VES) is a special public-key encryption scheme for encrypting *verifiable values*. A *verifiable value* is a number that satisfies a mathematical formula, such as a digital signature or a discrete logarithm of a given element etc. A VES can be used to encrypt such a *verifiable value*, meanwhile the encrypter can convince others that the encrypted value indeed meets the requirement (satisfies a mathematical formula) without disclosing the value.

A little bit more formally, let's assume that x is a discrete logarithm of G with base g, i.e., $G = g^x$. By a VES, an encrypter can encrypt x by DA's public key K_P. Denote the ciphertext by $C = K_P(x)$. The encrypter can prove to others that C is indeed the encryption of x without disclosing x. If the proof is passed, the value y obtained by decrypting C with DA's private key must satisfy $G = g^y$.

A VES of digital signature is similar. Let s be a digital signature on message m under verification key K_V. Then s passes a signature verification algorithm **veri**, i.e., $\mathbf{veri}(s, m, K_V) = yes$. By a VES, an encrypter can encrypt s with DA's public key K_P (denoting the ciphertext by $C = K_P(s)$), and prove to others that C is indeed the encryption of s under public key K_P, without disclosing s. If the proof is passed, the value y obtained by decrypting C with DA's private key must satisfy $\mathbf{veri}(y, m, V_K) = yes$.

A VES for discrete logarithm can be easily converted into a VES for digital signature. There have been several research papers proposing various VES for digital signatures. We will give a brief review of them later.

So far VESs have only been used to encrypt *verifiable values* with certain homomorphic verifiability, such as a discrete logarithm, an e-th root, or a signature. No VES of pre-image of a general one-way function, say a hash function, has ever been proposed. It seems to be a very hard problem.

3 An Analogy of VES and Its Applications

When explaining PKI to a layman, we can analogize a PKC with a box that has a lock embedded on it. The box can be left open with key removed. Everyone can lock the box but only the key owner can open it. For a similar intuitive illustration, we can analogize a VES with a glass box with a lock. The difference of a VES from an ordinary PKC is that everyone can "see" what is actually locked inside the box. But no one can get it except for the (private) key owner.

The VES of digital signature has great potential in the application when digital signatures are practically prevalent. One example is for the online auction where digital money is used. A digital money is a digital signature on a message of a certain format. The signature itself is the precious value while the format of the message should be public and in consistence with some standards.

Consider a situation of online auction where we

have a seller and some bidders. Every bidder bids a price. Now we have a dilemma whether the seller should collect all the digital money from all bidders. If yes, the seller is able to "escape" with all the money. If no, a malicious bidder can destroy the auction by bidding a very high price without actually giving the money. Everything may happen on Internet. By exploiting VES, such a dilemma can be perfectly resolved. We can ask each of the bidders to put his digital money into a glass box and lock the box by DA's public key. Then all bidders can give these boxes to the seller who is not able to open the box. But the seller can "see" that the money in the boxes are indeed what the bidders bid. The seller takes the bidder of highest price as the winner. In this case the seller cannot escape with all the money since he cannot open the boxes. On the other hand, no bidder can maliciously destroy the auction by over bidding since his money is in the box that can be opened by the DA. See Figure 1.

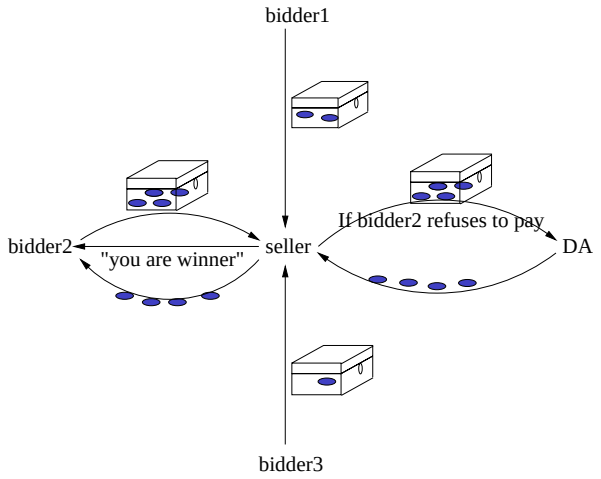

Figure 1: Solving the dilemma in online auction by VES

Of course here we must carefully specify the conditions on which the DA opens the box for the seller, in preventing any fraud from the seller.

The advantage of exploiting VES here is that the DA is off-line. If the bidders and the seller perform properly in the auction(which is the majority situation, we believe), the DA is not involved. This greatly decreases the communication burden of the DA.

One thing worth mentioning is that the seller is able to get the winner's digital money twice if he still goes to DA after receiving the money from the winner. But it does not matter since this is digital money. You can always copy digital money yourself.

Another possible application is online contract signing. Online contract signing is an important step in the B-to-B e-commerce. Before the real transaction, each party would like to obtain the other party's commitment to the transaction statement, i.e., the digital signature of a contract. But who should give his signature first is a problem. No one is willing to give out his signature first without any guarantee. VES can be used here as a guarantee where Party A gives his signature in the glass box to Party B. B can "see" that the stuff inside the box is indeed A's signature and B is convinced that the box is locked by DA's public key. Now he feels easier to give his signature to A. See Figure 2.

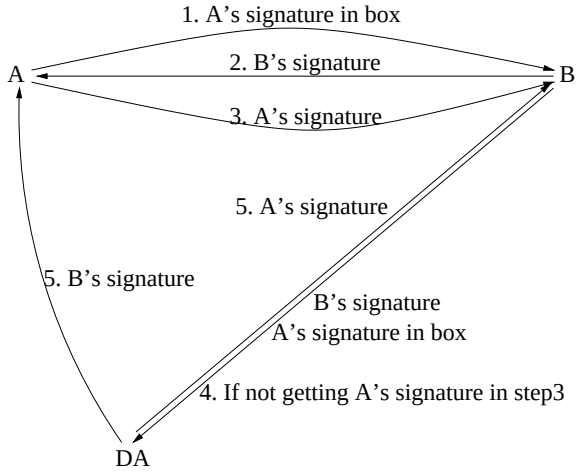

Figure 2: Guarantee of recoverability of A's signature by VES

Other applications include certified email where the sender wants to obtain receiver's signature as the receipt while the receiver wants to obtain the sender's signature as non-repudiation evidence.

In online shopping, fair exchange of the money and the e-goods or e-invoice is a desired feature, which can be realized by using VES and DA.

4 A Prospective Security Service

Cryptography has been intensively developed in the past two decades. A huge amount of research work in this area has been done. Although cryptography has lead to so many security technologies and products, it seems that market has mostly favored at two of them

so far, PKI and VPN. In security product exhibitions the most often seen products are PKI and VPN.

According to the study of Datamonitor Corporation, PKI market is expected to grow to 2.8 billion in the next three years.

year	US	Europe	Rest of World	Total
1999	158	58	16	232
2000	234	130	32	405
2003	627	545	191	1363

Table 1: PKI Product Revenues (figures in US millions)

year	US	Europe	Rest of World	Total
1999	195	90	23	308
2000	282	165	46	493
2003	610	542	223	1375

Table 2: PKI Implementation Service Revenues (figures in US millions)

Market analysis firm Datamonitor is even more optimistic, projecting $3.5 billion in total U.S. and European revenues from PKI products and services by 2003.

In the above survey figures, the PKI implementation service revenues is only for the CA part, i.e., the fees for issuing certificates. The typical valid period for a certificate is one year, which means that CA serves each client once per year.

Like CA, DA also provides service to its clients. Instead of issuing certificates, DA decrypts messages for clients. Unlike certificate, such service has no concept of valid period. Therefore, there is no limit on how many times of service a client request within one year. The service is provided whenever a client requests it. That is the reason why we say DA is potentially more profitable than CA.

DA's service can be generally described like this: When a client requests DA to decrypt a message C for her, DA decrypts C by his private key and checks whether the decrypted messages following certain format specified by this PKI system. The format should include two parts. The first part is the message M that the client wants. The second part is the conditions, only upon which DA gives M to the client. DA refuses to give M to the client if either the decrypted message does not follow the format "$M||$conditions", or the conditions are not satisfied. The detailed specifications on format and conditions should depend on the applications, similar to the attribute certificate defined in X.509 standard.

The example of conditions may be a time period within which DA returns M, or a requirement that the client should provide some digital objects, etc.

If M is a signature (the situation this paper mainly focuses on), VES should be used. In that case, the conditions are put into the "label" of VES. A "label" of VES is intuitively like a sticker put onto the glass box from inside before the box is locked. The "label" cannot be changed by anyone once the box is locked. But everyone can see the "label". More formal discussions will be presented in the last section.

Architecture of DA

DA consists of four main parts, RA(registration agent), DAA(DA agent), DPM(decryption policy managent), DAC(DA core). Such a classification is more from engineering viewpoint. This is similar to the classification within CA. Logically, DA, as well as CA, is a complete unit with one function.

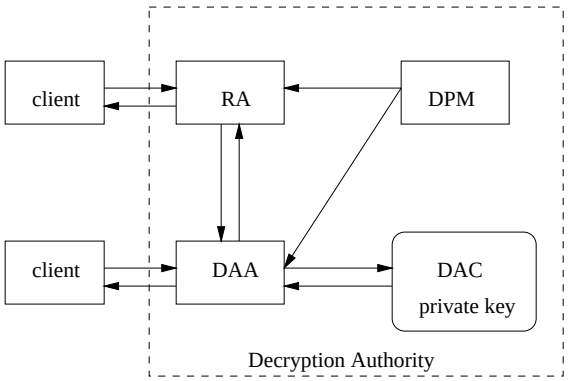

Figure 3: Outline of DA

Here DAC is the private key holder and the decryption implementor. DAC trusts DAA and decrypts whatever from DAA and sends the decryption outcome back to DAA. DAC should never disclose the private key, not even to DAA.

DPM is the decryption policy maker. It specifies what kind of clients can receive what kind of service under what condition. This party is most related to applications. It is more dynamic.

RA is the first place a client to go. It issues some id tokens or authentication keys to clients for their later accesses to DAA. RA is subject to the control from

DPM so that the clients can get proper previledges.

DAA is the party to decide whether to decrypt for a client. In one case, DAA has to ask the DAC to decrypt a ciphertext first, then check the "conditions". It gives the plaintext to the client only if the "conditions" are met. In the other case (using VES), DAA can determine whether the "conditions" are met before asking the DAC doing the decryption.

5 Related Research Work

As we mentioned earlier, if the encrypter who encrypts the message and the recipient of the ciphertext are not mutually trusted, the recipient must ask the encrypter to prove that what is encrypted is indeed that message. VES has to be exploited here. VES has been much studied recently. Many VESs have been proposed and applied to various situations, see [ASW98, Ate99, BDM98, Bao98, BT99, CM98, FO98, Sta96, YY98].

VES vs Zero-knowledge Proof

VES of digital signatures are different from those zero-knowledge proof protocols of owning a signature, like in [CD98, FO97, KP98, NBMV99]. In the zero-knowledge proof protocols a prover can prove that he owns a digital signature without any one being able to recover the signature, while in the VESs the prover not only prove he owns the signature but also must allow the owner of the decryption key being able to recover the signature. Therefore, in those zero-knowledge proof protocols the commitment function can be any one-way function, but in the VESs the commitment function must be a trapdoor one-way function. The second difference is that the zero-knowledge proof protocols aim at zero-knowledge proof while the VESs aim at the hardness of recovering the signature without the decryption key. Hence, disclosing half bits of the signature is regarded insecure in the zero-knowledge proof protocols, but still regarded secure in the former VESs.

Two Party VES vs Three-Party VES

VESs can be classified into two classes from the application angle. We call them *two-party* VESs, such as those in [BT99, CM98, FO98, Sta96, YY98], and *three-party* VESs, such as those in [ASW98, Ate99, BDM98, Bao98]. In two-party VESs, the final ciphertext receiver is the decrypter(the decryption key owner), and the decrypter need not return the decrypted value to any one else. The applications of the two-party VES include key escrow, group signature, verifiable secret sharing, etc. In the three-party VESs, the decrypter plays a role of a trusted third party (TTP). The TTP must return the decrypted value to the party who legally requests for the TTP's decryption (In our proposal, a DA from PKI plays this role of TTP). The applications include fair exchange of digital signatures, on-line auction, electronic payment systems, etc. Therefore, three-party VESs have one more requirement than two-party VESs, that is, resisting chosen ciphertext attack. From this viewpoint we say that a secure three-party VES must be a secure two-party VES, the converse direction does not necessarily hold.

Techniques to Construct VESs

In [ASW98, Sta96, YY98], the VESs are constructed by cut-and-choose method. Such a method brings comparatively large complexity both in computation and in message size. In [ASW98], the construction is very general that can combine any public-key scheme with any digital signature scheme. Actually, it presents a VES of pre-images of any homomorphic one-way function and an arbitrary PKC. In [Ate99, Bao98, BT99, FO98], VESs are constructed with methods different from cut-and-choose. The VES in [Bao98] is the most efficient one both in computation and in message size among all the VESs of discrete logarithms. Its security proof is similar to that of [PS96]. By the same method of [Bao98], the new public key scheme from Paillier [Pai99] can be converted into an efficient VES for discrete logarithm. [Ate99] presents a batch of efficient VESs of signatures. The VESs in [BT99, FO98] have smaller complexity in message size than that of [ASW98], but have similar or even larger computation complexity except for very small RSA exponent e, say, $e = 3$ or 5.

Advantage of VES in Applications

In a three-party VES, the decryption key owner is usually a TTP. The application may be a fair exchange protocol where the TTP is made off-line by exploiting VES. Detailed studies can be found in [ASW98] and [BDM98]. The advantage of off-line TTP is that when the two exchanging parties are honest, the TTP is not involved. Therefore, for majority of situations, the TTP seems not exist. And the fair exchange protocols using VES have much smaller number of communication rounds, compared with the protocols of [BGMR90, EGL85].

VES was applied to the multi-party fair exchange in [BDNV99] that has advantage of off-line TNP over the previous multi-party fair exchange protocols.

VESs has been applied to other situations, for example fair electronic payment systems in the same principle as in [BF98].

6 Labeled VES

Formal Description of a Three-Party VES

Let θ be a homomorphic one-way function. Encrypter encrypts a secret x to the ciphertext C by Decrypter's public key K_P and generates a certificate $Cert$ for proving that C is indeed the encryption of x such that $\theta(x) = X$. Encrypter gives $(C, Cert, X)$ to Verifier. Verifier verifies $(C, Cert, X, K_P)$. If the verification is passed, then the decryption result of C (by the private key of K_P), denoted by y, must satisfy $\theta(y) = X$.

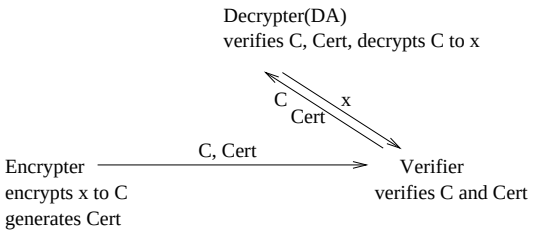

Figure 4: Three-party VES

Security Requirements of VES

1. Decrypter is always supposed to be honest(Decrypter plays the role of TTP in some applications). The security for Decrypter is that his private key won't be disclosed even if Encrypter and Verifier collude.

2. The security for Verifier is that if he is honest(following the scheme properly), he won't be fooled by Encrypter. In other words, once $(C, Cert, X, K_P)$ pass his verification, he can get the x such that $\theta(x) = X$ by the help from Decrypter.

3. The security for Encrypter is that x should not be disclosed to Verifier unless Verifier goes to Decrypter for decryption. That is, it is computationally hard to find x from $C, Cert, X$.

Convert VES to Labeled VES

In all the VESs, $Cert$ is a non-interactive proof in which the challenge is the value hashed from some messages (random oracle). We denote $Cert_A$ the non-interactive proof in which the challenge is the value hashed from the messages **and** A. Here A is called the label. In application of fair exchange, A may be a specification of the exchanging parties and values to be exchanged. In a labeled VES, the label A can never be changed or replaced.

7 Concluding Remarks

In this paper, we propose a new component for PKI, DA, which can greatly increase PKI implementation service revenue. DA decrypts important, valuable, sensitive messages for clients.

We basically classify DA's decryption into two classes. The first class is that the client does not know what is really encrypted. Here we assume that the client trusts the encrypter. Any public key encryption scheme can be used for the first class. The second class is that the client is able to check whether the encrypted message is what he wants, by exploiting VES. In this paper we focus more on the second class, since the first class is very simple from technology viewpoint.

In both first class and second class, the key point of implementation of DA is the specification of format. For example of the first class, the message being encrypted should be $M||condition$. In some application the *condition* could specify a *date* and a *recipient*, while DA only returns M to the *recipient* after that *date*. The specification of the format of *condition* must be carefully studied for any practice.

In the second class the technique is more complicated. The condition must be added to the label of a labeled VES. How to specify the format of label so that DA can automatically operate is very important in concrete applications.

References

[ASW97] N. Asokan, M. Schunter and M. Waidner, "Optimistic protocols for fair exchange", Proceedings of 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, pp. 6-17, April, 1997.

[ASW98] N. Asokan, V. Shoup and M. Waidner, "Optimistic fair exchange of digital signatures", in the Proceedings of Eurocrypt'98, LNCS, Springer-Verlag, 1998.

[Ate99] G. Ateniese, "Efficient protocols for verifiable encryption of digital signatures", Proceedings of the 6th ACM Conference on Computer and Communications Security, 1999.

[BDM98] F. Bao, R. H. Deng and W. Mao, "Efficient and practical fair exchange protocols with off-line TTP", in the Proceedings of the 1998 IEEE Symposium on Security and Privacy, IEEE Computer Press, Oakland, CA, 1998.

[Bao98] F. Bao, "An efficient verifiable encryption scheme for the encryption of discrete logarithms",

Proceedings of CARDIS'98, LNCS, Springer-Verlag.

[BDNV99] F. Bao, R. Deng, K. Nguyen, V. Varadharajan, "Multi-party fair exchange with an off-line TNP", Proceedings of DEXA'99 Workshop on Electronic Commerce and Security, Florence, Italy, 1999.

[BR93] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols", Proceedings of the First ACM Conference on Computer and Communications Security, 1993.

[BGMR90] M. Ben-Or, O. Goldreich, S. Micali and R. Rivest, "A fair protocol for signing contracts", IEEE Transactions on Information Theory, IT-36(1):40–46, January 1990.

[BF98] C. Boyd and E. Foo, "Off-line fair payment protocols using convertible signatures", Proceedings of Asiacrypt'98, LNCS 1514, Springer-Verlag, pp.271-285, 1998.

[BT99] F. Boudot and J. Traore, "Efficient publicly verifiable secret sharing schemes with fast or delayed recovery", Proceedings of ICICS'99, LNCS, Springer-Verlag , 1999.

[CM98] J. Camenisch and M. Michels, "A group signature scheme with improved efficiency", Proceedings of Asiacrypt'98, LNCS 1514, Springer-Verlag, pp. 160-174, 1998.

[CGH98] R. Canetti, O. Goldreich, and S. Halevi, "The random oracle methodology, revisit", in the Proceedings of STOC'98.

[CD98] R. Cramer and I. Damgard, "Zero-knowledge proofs for finite field arithmetic or: can zero-knowledge be for free?", Proceedings of Crypto'98, LNCS, Springer-Verlag, 1998.

[EGL85] S. Even, O. Goldreich and A. Lempel, "A Randomized Protocol for Signing Contracts", CACM, Vol. 28, No. 6, pp.637-647, 1985.

[ElG85] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory, IT-31(4):469-472, 1985.

[FR97] M. K. Franklin and M. K. Reiter, "Fair exchange with a semi-trusted third party", Proceedings of the 4th ACM Conferences on Computer and Communications Security, pp. 1-5, April 1-4, 1997, Zurich, Switzerland.

[FO97] E. Fujisaki and T.Okamoto, "Stastical zero-knowledge protocols to prove Modular Polynomial Relations", Proceedings of Crypto'97, LNCS 1294, pp. 16-30.

[FO98] E. Fujisaki and T. Okamoto, "A practical and provable secure scheme for publicly verifiable secret sharing and its application", Proceedings of Eurocrypt'98, Springer-Verlag, pp. 32-46, 1998.

[Gir91] M. Girault, "Self-certified public keys", Proceedings of Eurocrypt'91, LNCS 547, Springer-Verlag, pp. 490-497, 1991.

[GQ88] L. C. Guillou and J. J. Quisquater, "A paradoxical identity-based signature scheme resulting from zero-knowledge", Advances in Cryptology - Crypto'88, LNCS 403, Springer-Verlag, pp. 216-231.

[KP98] J. Kilian and E. Petrank, "Identity Escrow", Proceedings of Crypto'98, LNCS, Springer-Verlag, 1998.

[NBMV99] Q. K. Nguyen, F. Bao, Y. Mu and V. Varadharajan, "Zero-knowledge proofs of possession of ElGamal-like digital signatures and it applications", Proceedings of ICICS'99, LNCS, Springer-Verlag , 1999.

[OU98] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring", Proceedings of Eurocrypt'98, LNCS, Springer-Verlag, 1998.

[Pai99] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes", Proceedings of Eurocrypt'99, LNCS, Springer-Verlag, pp. 223-238, 1999.

[PS96] D. Pointcheval and J, Stern, "Security proofs for signature schemes", Proceedings of Eurocrypt'96, LNCS 1070, Springer-Verlag, pp. 387-398, 1996.

[PS99] G. Poupard and J. Stern, "On the fly signatures based on factoring", Proceedings of the 6th ACM Conference on Computer and Communications Security, 1999.

[Sta96] M. Stadler, "Publicly verifiable secret sharing", Proceedings of Eurocrypt'96, LNCS 1070, Springer-Verlag, pp.190-199, 1996

[YY98] A. Young and M. Yung, "Auto-recoverable auto-certifiable cryptosystems", Proceedings of Eurocrypt'98, LNCS, Springer-Verlag, pp. 17-31, 1998.

A The Public Key Encryption Scheme

In this appendix, we give the mathematical description of two efficient (labeled) VESs for discrete logarithm based digital signatures, one for DSA and the other for Schnorr signature scheme. The two VESs are very efficient with the computational cost of each VES just about twice that of the corresponding signature scheme. First let us look at the public key encryption we are using.

Consider Okamoto-Uchiyama trapdoor one-way function [OU98] with the following parameters:

• Private key: two primes p, q. Here p is recommended to have 350 bits and q 324 bits.
• Public key: $n = p^2q$ and g.

Here $g \leq n$ and $g \bmod p^2$ is a primitive element of $\mathbf{Z}^*_{p^2}$. Note that a randomly chosen g from $\mathbf{Z}_n$ satisfies the above requirement except for a negligible probability $1/p$.

Plaintext: $m \in \{0,1\}^{|p|-1}$
Encryption: $C = g^m \bmod n$
Decryption: $m = \frac{(C^{p-1} \bmod p^2)-1}{(g^{p-1} \bmod p^2)-1} \bmod p$

The correctness of the decryption:
Since $g^{p(p-1)} = 1 \bmod p^2$, we have

$$g^{p-1} = 1 \bmod p$$

Let

$$(g^{p-1} \bmod p^2) = lp + 1$$

then

$$C^{p-1} = (lp+1)^m = mlp + 1 \bmod p^2$$

Remarks on the OU trapdoor one-way function

- The security of the one-way function is based on the difficulty of factorization of $n = p^2q$.
- The above scheme cannot be taken as a public key encryption scheme since it is fragile to a chosen ciphertext attack: if we encrypt a message M larger than p and decrypt the ciphertext to get m, we can obtain p with large probability since $p = \gcd(n, M - m)$.
- The scheme can be easily changed to a public key encryption scheme that is semantically secure and resisting any chosen ciphertext attack. See [OU98].
- In our scheme, we use OU trapdoor one-way function instead of OU public key encryption. In our application, semantic security is not necessary since what being encrypted in VES is a discrete logarithm of a known value. That discrete logarithm is usually randomly chosen in the signature generation.
- We need to make it resisting chosen ciphertext attack since in our application the TTP should decrypt for others in resolution. What we exploit here is the so-called know-plaintext property.

B Labeled VES for Discrete Log

Besides the parameters in A, we have P, Q and G where P and Q are two primes such that Q is a factor of $P - 1$, and G is an element of $\mathbf{Z}^*_P$ of order Q. The recommended sizes of P and Q are 1024-bit and 160-bit, respectively. Let h be a one-way hash function of $\{0,1\}^* \to \{0,1\}^{160}$ used as Fiat-Shamir heuristics.

In the VES for DisLog, we have three parties

• TTP: who generates p, q, n, g etc, for OU scheme. n, g are public.
• Prover: who knows a secret $x < Q$ such that $G^x = Y \bmod P$, and encrypts x by the TTP's public key. *Label* is a binary string from Prover.
• Verifier: who verifies that the ciphertext from Prover is indeed the encryption of x.

Labeled VES for Digital Signatures

Prover:

compute $C = g^x \bmod n$ (encryption of x),

randomly choose $w \in \{0,1\}^{380}$,

set $a = g^w \bmod n$, $A = G^w \bmod P$, $r = h(C, Y, a, A, Label)$, $c = w - xr$,

make sure $0 < c < 2^{380} - 2^{320}$ (otherwise repeat until getting the required values)

Prover sends to Verifier:

$C, Y, r, c, Label$

Verifier:

check $r = h(C, Y, g^cC^r, G^cY^r, Label)$ and $0 < c < 2^{380} - 2^{320}$.

If the check is correct, the (r, c) is a valid certificate for equivalence of dislog of Y and decryption of C, with label *Lable*. Later when necessary, Verifier may ask TTP to open C for him.

Verifier sends to TTP:

$C, Y, r, c, Label$

TTP:

check $r = h(C, Y, g^c C^r, G^c Y^r, Label)$ and $0 < c < 2^{380} - 2^{320}$. If OK, compute $X = \frac{(C^{p-1} \bmod p^2)-1}{(g^{p-1} \bmod p^2)-1} \bmod p$ and let $x = X$ if $X < p/2$ or $x = X - P$ if $X > p/2$.

TTP sends Verifier:

$x \bmod Q$

This scheme evolves from the one in [Bao98]. The principle of the scheme is the non-interactive zero-knowledge proof of equality of discrete logarithms in different groups where one of the groups has unknown order. The computation of exponents is conducted in integer ring $\mathbf{Z}$ instead of in a prime field $\mathbf{Z}_q$. This technique has also been adopted in [Ate99, BT99, CM98, FO98, PS99].

We will not give formal security proof to the scheme here. Instead, we give a sketch to highlight the critical points of the proof.

- Disclosing $c = w - xr$, $0 < c < 2^{380} - 2^{320}$ does not lead to disclosing any bit of x. This is because [0, $2^{380} - 2^{320}$] is a common range of c for every x. In other words, for $0 < c < 2^{380} - 2^{320}$, every x in [0, 2^{160}] is equally probable. The information of g^w does not help in finding x.

- $c < 2^{380}$ guarantees $|x| < 350 = |p|$. This is because in $c = w - xr$, r is a 160-bit value hashed from some messages including g^w. If $|x| \geq 350$, xr has 510 bits. To make c only 380 bits, w must be 510 bits with higher 130 bits the same as xr. It is probabilistically impossible to find such w and r, since r is again dependent of w. For random oracle assumption of h, this is like to require the 160-bit outcome of h have 130 bits matching a format specified by the input. Even for birthday paradox the security level is 2^{-65}.

- The formal security proof can be established following the procedure and method in [PS96].

- With overwhelming large probability, we also have $|x| < |p| - 1$. So if the decrypted value $X > p/2$, that means the original x must be negative. So the returned value is $x = X - p$.

The parameter sizes like 160, 350, 380 etc here are just for simplicity. They should be formalized in formal proof.

C VES for DSA and Schnorr Signature

VES for DSA

Let P, Q and G to be the same as in B. DSA can be described as follows.

- Private key: $x \in \mathbf{Z}_Q$
- Public key: $y = G^x \bmod P$
- Signing: given a message m, randomly choose $k \in \mathbf{Z}_Q$, set $R = G^k \bmod P$ and $s = (m + xR)/k \bmod Q$. The signature on m is the pair (R, s). Here m is supposed to be the message already hashed by a one-way hash function.
- Verifying: check whether $R^s = G^m y^R \bmod P$.

The encryption of the signature is performed only on s while leaving R plain. Knowing R will not make it any easier to fabricate a signature since finding s such that $R^s = G^m y^R \bmod P$ requires solving the discrete logarithm problem. Moreover, R itself contains no useful information since k is randomly chosen (therefore, R is like a random variable). Hence, in our VES for DSA, the ciphertext of the signature is (R, C), where $C = g^s \bmod n$.

VES for Schnorr Signature

The Schnorr signature scheme has the same environment setting as that of the DSA. We assume that the same P, Q, G as in B are used here.

- Private key: $x \in \mathbf{Z}_Q$
- Public key: $y = G^x \bmod P$
- Signing: randomly choose $k \in \mathbf{Z}_Q$, set $K = G^k$, $R = h(m, K)$ and $s = k + sR \bmod Q$. The signature on m is the pair (R, s). Here h is the one-way hash function as in B.
- Verifying: check whether $R = h(m, G^s y^{-R})$.

The encryption of the VES is performed only on s while leaving R plain. But here we need to introduce K where $K = G^s y^{-R}$. The ciphertext of the Schnorr signature is (K, R, C) where $C = g^s \bmod n$. The verification of (K, R, C) is to prove
$R = h(m, K)$ and
$G^s = K y^R$ and
$g^s = C$.