ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.10, October 2007, pp.2662−2668
DOI: 10.1360/jos182662

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于双线性对的Chameleon签名方案*

杜欣军[1+], 王 莹[1], 葛建华[2], 王育民[2]

[1](西安电子科技大学 计算机网络与信息安全教育部重点实验室,陕西 西安 710071)

[2](西安电子科技大学 ISN国家重点实验室,陕西 西安 710071)

Chameleon Signature from Bilinear Pairing

DU Xin-Jun[1+], WANG Ying[1], GE Jian-Hua[2], WANG Yu-Min[2]

[1](Key Laboratory of Computer Networks and Information Security, Xidian University, Xi'an 710071, China)

[2](Key Laboratory of Integrated Services Network, Xidian University, Xi'an 710071, China)

+ Corresponding author: Phn: +86-29-88204749, Fax: +86-29-88204749, E-mail: dxjwy2002@hotmail.com

Du XJ, Wang Y, Ge JH, Wang YM. Chameleon signature from bilinear pairing. *Journal of Software*, 2007, 18(10):2662−2668. http://www.jos.org.cn/1000-9825/18/2662.htm

Abstract: Chameleon signatures are non-interactive signatures based on a hash-and-sign paradigm, and similar in efficiency to regular signatures. The distinguishing characteristic of chameleon signatures is that they are non-transferable, with only the designated recipient capable of asserting its validity. This paper introduces a new chameleon hash function based on bilinear pairing and builds the ID-based chameleon signature scheme. Compared with the conventional chameleon hashing functions, the owner of a public hash key in the ID-based chameleon hashing scheme does not necessarily need to retrieve the associated secret key. The scheme enjoys all the attributes in the normal chameleon signature and the added characteristics of ID-based cryptography based on bilinear pairing.
Key words: digital signature; bilinear pairing; Chameleon hashing; Chameleon signature

摘 要: Chameleon 签名方案是一种利用 Hash-and-Sign 模式的非交互签名方案,并且具有不可转移性,只有指定的接收者才可以确信签名的有效性.利用双线性对提出了一种新的 Chameleon Hash 函数,并在此基础上构建了相应的基于身份的 Chameleon 签名方案.与传统的 Chameleon Hash 函数相比,该方案中的 Hash 函数公钥所有者无须获取相应私钥,除非它企图伪造签名.该方案不但具有通常 Chameleon 签名方案的所有特点,而且具有基于身份密码系统的诸多优点.
关键词: 数字签名;双线性对;Chameleon hashing; Chameleon 签名
中图法分类号: TP309 **文献标识码**: A

1 Introduction

The conventional digital signature can be validated by any party, but this may be undesirable in many business

* Supported by the Communication Security Foundation of China under Grant Nos.J641, 0130 (国家通信保密基金); the National Natural Science Foundation of China under Grant No.69931010 (国家自然科学基金)

Received 2003-10-13; Accepted 2006-04-27

and e-commerce situations. Previous work has dealt with the problem of bridging between the contradictory requirements of non-repudiation and controlled dissemination via the notion of *undeniable signatures*. The notion was introduced by Chaum and van Antwerpen[1] and followed by many research works[1−5]. The basic paradigm behind this type of signatures is that verification of signature requires the collaboration of signer, so that the latter can control to whom the signed document is being disclosed. The crucial requirement is non-transferable, i.e. A signature issued to a designated recipient cannot be validated by another party. To prevent the leaking of information, these protocols are based on zero-knowledge proofs and this adds to the complexity of the schemes relative to the regular digital signatures.

Chameleon signature schemes were introduced in Ref.[6] which is a much simple implementation of the notion of undeniable signatures. The main technical novelty of chameleon signatures is in departing from the zero-knowledge paradigm. Unlike undeniable signatures, which also provides non-repudiation and non-transferability, chameleon signatures are non-interactive protocols. More precisely, the signer can generate the chameleon signature without interacting with the designated recipient, and the latter will be able to verify the signature without interacting with the former. Similarly, if presented with a forged signature, the signer can deny its validity by revealing certain values. These values will revoke the original signature and the forged one simultaneously, and the revocation can be universally verified. In other words, the forged-signature denial protocol is also non-interactive. Chameleon signatures are based on the well established hash-and-sign paradigm, where a chameleon hash function is used to compute the cryptographic message digest. A chameleon hash function is a trapdoor one-way hash function.

In this paper, we present a new chameleon signature scheme using a chameleon hash function from bilinear pairing. The scheme enjoys all the attributes of the chameleon signature and the advantages of *ID*-based cryptography from bilinear pairing over the elliptic curve.

The rest of the paper is organized as follows: the next section briefly explains the bilinear pairing and the Decisional Hash Bilinear Diffie-Hellman (DHBD) assumption. Section 3 gives a detailed description of our chameleon signature scheme. In Section 4, a heuristic security analysis is presented. Section 5 concludes the paper.

2 Bilinear Maps and the Bilinear Diffie-Hellman Assumption

Let G_1 and G_2 be two cyclic groups of order q for some large prime q. G_1 is a cyclic additive group and G_2 is a cyclic multiplicative group. We assume that the discrete logarithm problems in both G_1 and G_2 are hard. Let $\hat{e}: G_1 \times G_1 \to G_2$ be a pairing which satisfies the following conditions:

(1) Bilinear: $\hat{e}(aP,bQ)=\hat{e}(P,Q)^{ab}$, for all $P,Q\in G_1$ and all $a,b\in Z_q^*$;

(2) Non-degenerate: there exists $P\in G_1$ and $Q\in G_1$ such that $\hat{e}(P,Q)\neq 1$;

(3) Computability: there is an efficient algorithm to compute $\hat{e}(P,Q)$ for all $P,Q\in G_1$.

We note that the Weil and Tate pairings associated with supersingular elliptic curves or abelian varieties can be modified to create such bilinear maps. We refer to Refs.[7−10] for more details.

BDH Parameter Generator: We say that a randomized algorithm IG be a BDH parameter generator if (1) IG takes a security parameter $0<k\in Z$, (2) IG runs in polynomial time in k, and (3) IG outputs the description of two groups G_1, G_2 and the description of a bilinear map $\hat{e}: G_1 \times G_1 \to G_2$ described above.

Decisional Hash Bilinear Diffie-Hellman (DHBDH) problem in $\langle G_1, G_2, \hat{e}\rangle$:

Instance: (P,aP,bP,cP,r) for some $a,b,c,r\in Z_q^*$ and a one way hash function $H: G_2 \to Z_q^*$.

Solution: Output *Yes* if $r=H(\hat{e}(P,P)^{abc}) \bmod q$ and output *No* otherwise.

The advantage of any probabilistic, polynomial time, 0/1-valued algorithm A in solving DHBDH problem in

$\langle G_1, G_2, \hat{e}\rangle$ is defined to be:

$$Adv_{\mathcal{A}}^{DHBDH} = |\Pr ob[\mathcal{A}(P, aP, bP, cP, r) = 1] - \Pr ob[\mathcal{A}(P, aP, bP, cP, H(\hat{e}(P,P)^{abc})) = 1]|, a,b,c,r \in_R Z_q^*.$$

DHBDH assumption: There exists no polynomial time algorithm, which can solve the DHBDH problem with non-negligible probability of success. In other words, for every probabilistic, polynomial time, 0/1-valued algorithm A, $Adv_{\mathcal{A}}^{BHBDH} < \frac{1}{m^{\ell}}$ for every fixed $\ell > 0$ and sufficiently large *m*.

3 Chameleon Signature Scheme from Bilinear Pairing

The chameleon signatures based on bilinear pairings apply a regular digital signature scheme (such as RSA or DSS) to a special type of hashing called ID-based Chameleon hash functions. The basic idea is to build the signature scheme in such a way that a signature provided by a signer *S* to a recipient *R* gives *R* the ability to gorge further signatures of *S* at will. Clearly this prevents *R* from proving the validity of *S*'s signature to a third party as he could have produced such a signature by himself.

3.1 *ID*-Based chameleon hashing

Here we present an ID-based chameleon hashing scheme from bilinear pairing and based on DHBDH assumption. We assume that all system users are identifiable by a bit-string easily derivable from public knowledge about the individual. Formally, an ID-based chameleon hashing scheme is defined by a family of efficiently computable algorithms: **Setup**, **Extract**, **Hash** and **Forge**.

Setup: A trusted party, Trusted Authorities (TA), works as follows:

Step 1. Run some BDH parameter generator IG on the input of a security parameter *k* to generate two prime order groups G_1, G_2 and the description of a bilinear map $\hat{e}: G_1 \times G_1 \to G_2$ described above. Choose an arbitrary generator$P \in G_1$.

Step 2. Pick a random $s \in Z_q^*$ and set $P_{pub}=sP$.

Step 3. Choose cryptographic hash functions: $H_1: \{0,1\}^* \to G_1^*$, $H_2: \{0,1\}^* \to Z_q^*$, $H: G_2 \to \{0,1\}^n$, for some *n*. The system public parameters are $params = \langle G_1, G_2, \hat{e}, n, P, P_{pub}, H_1, H_2, H\rangle$. The master-key is $s \in Z_q^*$.

Extract: A deterministic algorithm run by TA outputs the trapdoor information *B* associated to some identity. For a given string $ID \in \{0,1\}^*$, the algorithm does: (1) compute $Q_{ID} = H_1(ID) \in G_1^*$, and (2) set the trapdoor information $B=sQ_{ID}$.

Hash: A probabilistic algorithm that inputs the system public parameters *params*, an identity *ID*, a random $r \in_R G_1$ and a message *m*, outputs a hash value *h*. The algorithm is always run by the signer *S* and *ID* is the identity string of the recipient *R*. The algorithm does:

(1) $Q_{ID} = H_1(ID) \in G_1^*$.

(2) $h = Hash(params, ID, m, r) = H(\hat{e}(Q_{ID}, P_{pub})^{H_2(m)} \hat{e}(r, P))$.

Forge: A algorithm that inputs the system public parameters *params*, an identity string *ID*, the trapdoor information *B* associated with *ID*, a message *m'*, and a hash value *h* of a message *m*, outputs a random $r' \in_R G_1$ that corresponds to a valid computation of **Hash** for yielding the target value *h*.

The **Forge** algorithm is:

$$Forge(params, ID, B, m, r, h, m') = r' = H_2(m)B + r - H_2(m')B.$$

Not indeed that

$$\begin{aligned}Hash(params, ID, m', r') &= H(\hat{e}(Q_{ID}, P_{pub})^{H_2(m')}\hat{e}(r', P))\\ &= H(\hat{e}(sQ_{ID}, P)^{H_2(m')}\hat{e}(r', P))\\ &= H(\hat{e}(B, P)^{H_2(m')}\hat{e}(r', P))\\ &= H(\hat{e}(H_2(m')B, P)\hat{e}(r', P))\\ &= H(\hat{e}(H_2(m')B + r', P))\\ &= H(\hat{e}(H_2(m')B, H_2(m)B + r - H_2(m')B, P))\\ &= H(\hat{e}(H_2(m)B + r, P))\\ &= H(\hat{e}(Q_{ID}, P_{pub})^{H_2(m)}\hat{e}(r, P))\\ &= Hash(params, ID, m, r).\end{aligned}$$

3.2 Chameleon signature schemes

Here we present in some detail the chameleon signature scheme. The chameleon signature is generated by digitally signing a chameleon hash value of message. The digital signature scheme used here is some regular digital signature scheme. We start by describing the setting for chameleon signatures. The setting defines the players and the agreement upon functions and keys.

Players: Signer *S* and recipient *R*. In addition we shall refer to a judge *J* who represents a party in charge of settling disputes between *S* and *R*, and with whom *S* is assumed to collaborate.

Functions: The players agree on:

- A digital signature scheme (e.g., RSA, DSS) which defines a set of public and private keys associated with the signer and usual operations of signing denoted by *SIGN*, and verification denoted by *VERIFY*.
- A chameleon hashing function *Hash* which defines a set of public and private keys associated with the owner of the hash function. This function has been described in Section 3.1.

Key:

- The signer *S* has a public and private signature key which corresponds to the agreed on signature scheme denoted by VK_S and SK_S respectively.
- The recipient *R* has a public and private key as required by the agreement upon chameleon hashing scheme. Here the public key is *R*'s identifier ID and the private key is the trapdoor information $B=sQ_{ID}$ (Section 3.1).

ID-Based Chameleon Signature Generation-*CHAM-SIG*:

Input of Signer: Message *m*

Private signing key of *S*, SK_S

R's chameleon hashing public key, i.e. *R*'s identifier *ID*

1. Generate the chameleon hash of *m* by choosing a random $r \in_R G_1$ and compute
$$hash = Hash(params, ID, m, r) = H(\hat{e}(Q_{ID}, P_{pub})^{H_2(m)}\hat{e}(r, P))$$
2. Set $sig = SIGN_{SK_S}(hash, ID)$.
3. The signature on the message *m* consists of $SIG(m)=(m,r,sig)$.

ID-Based Chameleon Signature Verification-*CHAM-VER*:

Input: $SIG(m)=(m,r,sig)$;

Public verification key of *S*: VK_S

R's chameleon hashing private key, i.e. *R*'s trapdoor information *B*

1. Compute $hash=Hash(params,ID,m,r)$
2. output= $\begin{cases} proper, & VERIFY_{VK_S}((hash, ID), sig) = valid \\ improper, & \text{Otherwise} \end{cases}$

Dispute:

In case of a dispute on the validity of a signature, R can turn to an authorized judge J. J gets from R a tripe

$$SIG(\hat{m})=(\hat{m},\hat{r},\hat{sig}).$$

1. J applies the above *CHAM-VER* function. If this verification fails, then the alleged signature is rejected by J. Otherwise,
2. J summons the signer to deny/accept the claim. J sends to S the triple $SIG(\hat{m})$.
3. If S wants to claim that the signature is invalid he will need to provide a collision in the chameleon hash function. Otherwise, S simply confirms to judge this fact.

The following is the process that S generates collision in the hash function.

Generate Collision:

Input: a forgery $SIG(m')=(m',r',sig)$

1. S retrieves the original value m, r used to compute sig. It holds that
   $Hash(params,ID,m,r)=Hash(params,ID,m',r')$, while $m\neq m'$.
2. S computes $B=\dfrac{r'-r}{H_2(m)-H_2(m')}$.
3. S chooses any message $\overline{m}$ and computes $\overline{r}=\dfrac{H_2(m)-H_2(\overline{m})}{H_2(m)-H_2(m')}(r'-r)+r$.
4. Output $(\overline{m},\overline{r})$.

With the triple $SIG(\overline{m})=(\overline{m},\overline{r},sig)$, S can convince the judge to reject the false signature

$$SIG(m')=(m',r',sig).$$

4 Security Analysis

First we summarize the security properties that we require from a chameleon signature scheme.

- **Unforgeability**. No third party can produce an (R,S)-proper signature not previously generated by the signer.
- **Non-transferability**. Except for the signer himself, no one can prove to another party that the signer produced a given signature.
- **Denial**. The signer can convince the judge to reject a forgery signature.
- **Non-repudiation**. The signer cannot convince the judge to reject a signature produced by him.
- **Exposure freeness**. A chameleon signature scheme is exposure free if the signer can deny a false signature without exposing any other message actually signed by him.

If the above properties are satisfied, the chameleon signature from bilinear pairing is a secure chameleon signature scheme.

Theorem 1. Assuming a secure digital signature scheme and the hardness of DHBDH problem, the chameleon signature from bilinear pairing is secure.

Proof: Unforgeability. No third party can produce an (R,S)-proper $SIG(m)=(m,r,sig)$ not previously generated by the signer, as this requires either to break the underlying regular digital signature scheme, or to find collision of the ID-based chameleon hash function which, in turn, implies the settling of the DHBDH problem. The recipient also cannot produce a signature with a new component sig, as this requires to break the regular digital signature.

Denial. From the **Generate Collision** process (Section 3.2), we can see if the signature is false, the signer can convince the judge to reject the forgery signature by generating collision in the hash function.

Non-transferability. Given a signature $SIG(m)=(m,r,sig)$ generated by S for R, the recipient cannot convince a third party of its validity. From the **Forge** produced in the ID-based chameleon hashing scheme (Section 3.1), we

can see that for every possible message m', R can computer a value $r'=H_2(m)B+r-H_2(m')B$ such that $Hash(params,ID,m',r')=Hash(params,ID,m,r)$. Thus, (m',r',sig) is an (R,S)-proper signature. Furthermore, since for every possible message m' there exists exactly one value r' that produces a proper triple (m',r',sig), then nothing is learned about the value of m from seeing the signature string sig. Thus non-transferability is achieved unconditionally, i.e. in the information theoretic sense.

Non-repudiation. Given a $SIG(m)=(m,r,sig)$ generated by the signer S, S can not generate another (R,S)-triple $SIG(m')=(m',r',sig)$ for $m\neq m'$, as this would be equivalent to finding a collision of ID-based chameleon hash function , which we assume to be infeasible by the hardness of the DHBDH problem.

Exposure freeness. From the **Dispute** process (Section 3.2), we can see the signer utilizes the false signature and the original signature to produce false signature for any message with the same component *sig* without leaking anything about the original signature.

5 Conclusion

A chameleon signature from bilinear pairing is presented in this paper, which enjoys all the attributes in the normal chameleon signature. Additionally, it owns the characteristics of ID-based cryptography based on bilinear pairing. For example, a signer can sign a message to an intended recipient without having to first retrieve the recipient's certificate, because everyone who knows the identifier of a recipient can produce the public key of the corresponding ID-based chameleon hash function. The signer can use a different public key for each transaction with a recipient without having to retrieve a new certificate. Only the trusted third party can extract the trapdoor information and the recipient does not have to know the trapdoor information unless he wants to forge the signature.

Acknowledgement The authors would like to thank anonymous reviewers for their valuable comments.

References:

[1] Chaum D, Antwerpen H. Undeniable signatures. In: Brassard G. Advances in Cryptology-CRYPTO'89. Springer-Verlag, 1991. 212−216.

[2] Boyar J, Chaum D, Damgard IB, Pedersen TP. Convertible undeniable signatures. In: Menezes A, Vanstone SA, eds. Advances in Cryptology-CRYPTO'90. Springer-Verlag, 1990. 189−205.

[3] Chaum D. Zero-Knowledge undeniable signature. In: Damgard I, ed. Advances in Cryptology-EURPCRYPT'90. Aarhus: Springer-Verlag, 1990. 458−464.

[4] Chaum D, van Heijst E, Pfitzmann B. Cryptographically strong undeniable signatures, unconditionally secure for the signer. In: Feigenbaum J, ed. Advances in Cryptology-CRYPTO'91. Springer-Verlag, 1990. 470−484.

[5] van Heijst E, Pedersen T. How to make efficient fail-stop signatures. In: Rueppel RA, ed. Advances in Cryptology-EURPCRYPT'92. Balatonfured: Springer-Verlag, 1993. 366−377.

[6] Krawczyk H, Rabin T. Chameleon signature. In: Proc. of the Network and Distributed System Security Symp. (NDSS 2000). The Internet Society, 2000. 143−154.

[7] Boneh D, Franklin M. Identity-Based encryption from the Weil pairing. In: Kilian J, ed. Advances in Cryptology-CRYPTO 2001. Springer-Verlag, 2001. 213−229.

[8] Boneh D, Lynn B, Shacham H. Short signatures from the Weil pairing. In: Boyd C, ed. Advances in Cryptology-ASIACRYPT 2001. Gold Coast: Springer-Verlag, 2001. 514−532.

[9] Gentry C, Silverberg A. Hierarchical ID-based cryptography. In: Zheng YL, ed. Advances in Cryptology-ASIACRYPT 2002. Queenstown: Springer-Verlag, 2002. 213−229.

[10] Horwitz J, Lynn B. Toward hierarchical identity-based encryption. In: Knudsen LR, ed. Advances in Cryptology-EURPCRYPT 2002. Amsterdam: Springer-Verlag, 2002. 466−481.

DU Xin-Jun was born in 1974. He is a Lecturer at the Xidian University. His current research areas are cryptology, digital communication, etc.

GE Jian-Hua was born in 1961. He is a professor at the Xidian University. His current research areas are cryptology, communication, etc.

WANG Ying was born in 1974. She is a Ph.D. candidate at the Xidian University. Her current research areas are cryptology, communication, etc.

WANG Yu-Min was born in 1936. He is a professor at the Xidian University. His current research areas are cryptology, coding theory, etc.

敬 告 作 者

《软件学报》创刊以来,蒙国内外学术界厚爱,收到许多高质量的稿件,其中不少在发表后读者反映良好,认为本刊保持了较高的学术水平.但也有一些稿件因不符合本刊的要求而未能通过审稿.为了帮助广大作者尽快地把他们的优秀研究成果发表在我刊上,特此列举一些审稿过程中经常遇到的问题,请作者投稿时尽量予以避免,以利大作的发表.

1. 读书偶有所得,即匆忙成文,未曾注意该领域或该研究课题国内外近年来的发展情况,不引用和不比较最近文献中的同类结果,有的甚至完全不列参考文献.

2. 做了一个软件系统,详尽描述该系统的各个方面,如像工作报告,但采用的基本上是成熟技术,未与国内外同类系统比较,没有指出该系统在技术上哪几点比别人先进,为什么先进.一般来说,技术上没有创新的软件系统是没有发表价值的.

3. 提出一个新的算法,认为该算法优越,但既未从数学上证明比现有的其他算法好(例如降低复杂性),也没有用实验数据来进行对比,难以令人信服.

4. 提出一个大型软件系统的总体设想,但很粗糙,而且还没有(哪怕是部分的)实现,很难证明该设想是现实的、可行的、先进的.

5. 介绍一个现有的软件开发方法,或一个现有软件产品的结构(非作者本人开发,往往是引进的,或公司产品),甚至某一软件的使用方法.本刊不登载高级科普文章,不支持在论文中引进广告色彩.

6. 提出对软件开发或软件产业的某种观点,泛泛而论,技术含量少.本刊目前暂不开办软件论坛,只发表学术文章,但也欢迎材料丰富,反映现代软件理论或技术发展,并含有作者精辟见解的某一领域的综述文章.

7. 介绍作者做的把软件技术应用于某个领域的工作,但其中软件技术含量太少,甚至微不足道,大部分内容是其他专业领域的技术细节,这类文章宜改投其他专业刊物.

8. 其主要内容已经在其他正式学术刊物上或在正式出版物中发表过的文章,一稿多投的文章,经退稿后未作本质修改换名重投的文章.

本刊热情欢迎国内外科技界对《软件学报》踊跃投稿.为了和大家一起办好本刊,特提出以上各点敬告作者.并且欢迎广大作者和读者对本刊的各个方面,尤其是对论文的质量多多提出批评建议.