基于 DAA 的可信双向匿名认证密钥协商协议①

Trusted Authenticated Key Agreement Protocol with Bilateral Anonymity Based on DAA

关晨至 (江西教育学院 理学分院 江西 南昌 330029)

石永革 (南昌大学 信息工程学院 江西 南昌 330031)

摘 要: 现有的匿名认证密钥协商协议，无法实现通信双方相互之间匿名认证，只能实现单向匿名认证，而某些应用场合往往需要实现双向匿名认证。为此，基于椭圆曲线密码学和双线性映射的 DAA 协议，在可信平台上设计了一个提供双向匿名认证的密钥协商协议，使通信双方能够相互验证对方具有某种成员关系，又不暴露各自的真实身份。该协议适合于计算和存储资源有限的应用场合。

关键词: 认证密钥协商 可信双向匿名 DAA 双线性对

1 引言

在很多应用场合中，匿名性是一个重要的需求，以保证用户的身份信息不被暴露。对于认证密钥协商(Authenticated key agreement)协议，匿名性也有特殊的应用。在 Boyd 和 Park[1]与 Shoup[2]分别提出的匿名认证密钥协商协议中，假设两个实体 A 和 B 交换秘密消息，实现了在网络中除了 B 以外任何人都不知道 A 的身份，即 A 对其它人匿名。但在某些应用中，A 可能进一步希望包括 B 在内的任何人都不知道自己的身份，这就需要设计具有更强匿名性的协议，使得 A 和 B 通信时，B 可以确定 A 是一组用户中的一个，但无法获知 A 的确切身份，这特别适合于当 A 因为隐私而不能暴露自己的身份，同时 B 也只需确定 A 具有某种成员资格的场合。另外，现有的大多数匿名认证密钥协商协议只提供单向匿名性，但在某些应用中，通信双方希望相互认证时均不暴露自己的身份，从而需要提供双向匿名认证的功能。此外，在将来的普适计算环境中存在大量小型嵌入式设备，其计算能力和存储资源有限，传统的 DAA 协议由于利用 RSA 和离散对数实现，使得其运算量和密钥长度都比较大，不适合于计算和存储资源有限的应用场合。

随着网络技术的不断发展，可信计算正成为目前安全研究的热点。在可信计算中，隐私保护是可信系统的必要功能，为了实现可信平台模块(trusted platform module, TPM)之间相互认证又能保持匿名性以防止用户的行为被跟踪，可信计算组织发布的方案包括 TPM v1.1 的 Privacy CA 方案[3]和 TPM v1.2 的直接匿名证言(direct anonymous attestation, DAA)方案[4]。在文献[5]中，Ernie Brickell 等人提出了新的基于椭圆曲线密码学和双线性映射的 DAA 方案，与传统 DAA 相比有效地缩短了密钥和签名的长度，并保持了原来的安全性，更适合于计算和存储资源有限的应用场合。本文基于椭圆曲线密码学和双线性映射的 DAA 方案，在可信平台上构造了一个密钥协商协议，实现了双向匿名认证，使通信双方能够认证对方与自己具有某种成员关系，但无法探知对方的具体身份。

2 DAA协议

2.1 传统 DAA 协议

DAA 协议中有四种实体：DAA issuer、TPM、host 和 verifier。其中，TPM 和 host 构成可信平台

① 收稿时间:2009-06-02

并共同组成 DAA signer，issuer 作为可信第三方，协议的最终目标是实现 verifier 对 signer 的匿名认证。整个协议由五个算法组成：setup、join、sign、verify 和 link。issuer 运行 setup 初始化自己的公私钥对和一些安全参数；signer 与 issuer 联合运行 join 算法，其间 issuer 验证 signer 的合法性，然后向 signer 颁发 DAA 证书；signer 运行 sign 算法，利用自己的私钥和 DAA 证书，生成 DAA 签名，该签名作为 signer 拥有 DAA 签名和私钥的知识证明；signer 将 DAA 签名和相关消息发送给 verifier，verifier 运行 verify 算法，通过 DAA 签名验证 signer 是否拥有 issuer 颁发的 DAA 证书及合法私钥，但无法判断 signer 的真实身份；verifier 可以运行 link 算法来判断两个 DAA 签是否由同一个 signer 生成。

2.2 双线性对 DAA 协议

基于双线性对的 DAA 协议规程与传统 DAA 协议相同，保留了原有的实体和协议算法。在协议算法实现上，用基于椭圆曲线密码学和双线性对取代了基于 RSA 和离散对数密码系统，从而大大降低了对存储容量的要求。例如，原 DAA 协议中私钥和 DAA 签名的长度分别为 670 字节和 2800 字节左右，而在新的 DAA 协议中分别为大约 213 字节和 521 字节，从而更适合于计算和存储资源有限的应用场合。

3 可信双向匿名认证密钥协商协议

在该协议中，有 initiator、responder 和 CA 三种实体。initiator 是发起密钥协商请求的一方；responder 是与 initiator 进行密钥协商的另一方，双方都是由包含 TPM 的主机构成的可信平台；CA 是第三方，它在 initiator 和 responder 之间建立某种成员关系。协议分成初始化和认证密钥协商两个阶段，其中初始化只需进行一次，之后每次密钥协商时不再进行。

3.1 初始化

3.1.1 CA 运行 DAA setup 算法

(1) 选择 G_1、G_2、G_1 的生成元 g_1，G_2 的生成元 g_2，G_1 是阶为素数 q 的椭圆曲线上的加法循环群，G_2 是阶为 q 的有限域上的乘法循环群；选择双线性对 e，满足 e：$G_1 \times G_1 \rightarrow G_2$ 且 $e(P, P) \neq 1_{G2}$。

(2) 选择 $x \in {}^{\mathbb{R}}\mathbb{Z}_q$，$y \in {}^{\mathbb{R}}\mathbb{Z}_q$，私钥 sk=(x, y)。

(3) 计算 $X=xg_1$，$Y=yg_2$，公钥 pk=(q，g_1，G_1，g_2，G_2，e，X，Y)。

(4) 构造两个哈希函数，$H: \{0,1\}^* \rightarrow \{0,1\}^h$，$H_G: \{0,1\}^* \times \{0,1\}^* \times G_2 \rightarrow \{0,1\}^k$，其中 h 为 Fiat-Shamir 启发式方法[6]要求的长度，k 为会话密钥的长度。

CA 公布 pk、H、H_G。接下来 initiator 和 responder 分别与 CA 运行 DAA 的 join 算法。由于二者运行步骤相同，因此以下统一进行描述。双方各自申请 DAA 证书

TPM 内部生成私钥 f，计算公钥 $F=fg_1$，选择随机数 $r_f \in {}^{\mathbb{R}}\mathbb{Z}_q$，计算 $T = r_f g_1$。

TPM 把 F 和 T 输出到外部主机；主机向 CA 获取一个随机串 $n_I \in \{0,1\}^h$，计算 $c_h = H(q \| g_1 \| g_2 \| X \| Y \| F \| T \| n_I)$，输入 TPM。

TPM 生成随机串 $n_T \in \{0,1\}^{\phi}$，计算 $c = H(c_h \| n_T)$，$s_f = r_f + c \cdot f \bmod q$，把 (F, c, s_f, n_T) 通过主机传送给 CA。

CA 验证申请并颁发证书

(1) CA 检查 F 的合法性，如果 F 在黑名单中或者不符合某些安全策略，则终止协议。

(2) 若 F 合法，计算 $T = g^{s_f} F^{-c}$，然后验证等式 $c = H(H(q \| g_1 \| g_2 \| X \| Y \| F \| T \| n_I) \| n_T)$ 是否成立，若不成立，CA 终止协议。

(3) CA 选择 $r \in {}^{\mathbb{R}}\mathbb{Z}_q$，计算 $a = rg_1, b = ya$，$c = xa + (rxy)F$。(a,b,c) 作为 TPM 的 DAA 证书，发送给 TPM 和主机。

3.2 认证密钥协商

3.2.1 initiator 发起会话

(1) initiator 的主机选择 $B \in {}_R G_2$，输入到 TPM，TPM 验证 $B \in G_2$，计算 $K_i = B^{f_i}$，K_i 输出到主机。

(2) 主机选择 $r, r' \in {}^{\mathbb{R}}\mathbb{Z}_q$，计算 $a_i' = a_i^{r'}$，$v_x = e(X, a_i'), b_i' = b_i^{r'}$，$v_{xy} = e(X, b_i'), c_i' = c_i^{r' r^{-1}}, v_s = e(g_1, c_i')$。

主机输入 v_{xy} 到 TPM，后者验证 $v_{xy} \in G_2$。

(3) 主机和 TPM 联合计算关于 DAA 证书的签名证明。

主机选择 $r_r \in {}^{\mathbb{R}}\mathbb{Z}_q$，计算 $T_{1t} = v_s^{r_r}$，$c_h = H(q \| g_1 \| g_2 \| X \| Y \| a_i' \| b_i' \| c_i' \| v_x \| v_{xy} \| v_s \| B \| K_i \| TS_1)$。

TS_1 是时间戳，主机把 c_h 和 T_{1t} 输入 TPM。

TPM 选择 $r_f \in {}^{\mathbb{R}}\mathbb{Z}_q$，随机串 $n_T \in \{0,1\}^{\phi}$，计算

$T_1 = T_{1t} v_{xy}^{-r_f}$，$T_2 = B^{r_f}$，$S = H(c_h \| T_1 \| T_2 \| n_T \| Alias_i)$，$Alias_i$ 是 initiator 的别名，$s_f = r_f + S \cdot f \bmod q$。TPM 将 S，$s_f$，$n_T$ 输出到主机。

主机计算 $s_r = r_r + S \cdot r \bmod q$。

(4) initiator 将 $Alias_i$、TS_1、n_T 和 DAA 签名 $\sigma = (B, K_i, a_i', b_i', c_i', S, s_r, s_f)$ 发送给 responder。

initiator→responder：$Alias_i \| TS_1 \| n_T \| \sigma$

3.2.2 responder 认证 initiator

(1) responder 匿名认证 $Alias_i$：首先验证 $K_i \in G_2$，$e(a_i', Y) = e(g_1, b_i')$ 是否成立，然后计算 $v_x = e(X, a_i'), v_{xy} = e(X, b_i'), v_s = e(g_1, c_i')$，$T_1 = v_s^{s_r} v_{xy}^{-s_f} v_x^{-S}$，$T_2 = B^{s_f} K_i^{-S}$，最后验证等式 $S = H(H(q \| g_1 \| g_2 \| X \| Y \| a_i' \| b_i' \| c_i' \| v_x \| v_{xy} \| v_s \| B \| K_i \| TS_1) \| T_1 \| T_2 \| n_T \| Alias_i)$ 是否成立，若成立，则匿名认证成功。

(2) responder 响应请求：responder 选择与 initiator 签名中相同的 B，然后通过与 initiator 相同的步骤，对 $Alias_r$、TS_2 和 n_T' 生成 DAA 签名 $\sigma' = (B, K_r, a_r', b_r', c_r', S', s_r', s_f')$，发送给 initiator，其中 $Alias_r$ 是 responder 的别名，TS_2 是时间戳，n_T' 是 responder 的 TPM 生成的随机串。

responder → initiator：$Alias_r \| TS_2 \| n_T' \| \sigma'$

(3) responder 计算会话密钥：$skey_r = H_G(Alias_r, Alias_i, T_2^{r_f'})$，其中 T_2 是验证 initiator 的签名时的计算值，r_f' 是 responder 生成 DAA 签名时 TPM 选择的随机值。

3.2.3 initiator 认证 responder

(1) initiator 收到消息后，匿名认证 $Alias_r$。首先验证 $K_r \in G_2$，$e(a_r', Y) = e(g_1, b_r')$ 是否成立，然后计算 $v_x' = e(X, a_r'), v_{xy}' = e(X, b_r'), v_s' = e(g_1, c_r')$，$T_1' = v_s^{s_r'} v_{xy}^{-s_f'} v_x^{-S'}$，$T_2' = B^{s_f'} K_r^{-S'}$，最后验证等式 $S = H(H(q \| g_1 \| g_2 \| X \| Y \| a_r' \| b_r' \| c_r' \| v_x'$ 是否成立，若成立，则匿名认证成功。

(2) initiator 计算会话密钥：$skey_i = H_G(Alias_i, Alias_r, T_2'^{r_f})$，其中 T_2' 是验证 responder 的签名时的计算值，r_f 是 initiator 生成 DAA 签名时 TPM 选择的随机值。

3.3 协议正确性证明

若双方互相认证成功，则 initiator 和 responder 各自计算的会话密钥是相等的，因为

$$T_2 = B^{r_f}, T_2' = B^{r_f'}$$

$$skey_r = H_G(Alias_r, Alias_i, T_2^{r_f'}) = H_G(Alias_r, Alias_i, B^{r_f r_f'})$$

$$skey_i = H_G(Alias_i, Alias_r, T_2'^{r_f}) = H_G(Alias_i, Alias_r, B^{r_f' r_f})$$

4 安全性分析

(1) 双向匿名认证

initiator 和 responder 在密钥协商的初始化阶段，分别向 CA 申请 DAA 证书，成为 CA 的成员。在认证密钥协商阶段，initiator 和 responder 互相交换对各自别名、时间戳和随机串的 DAA 签名，然后双方分别验证签名，如果验证成功就表示对方拥有合法的 DAA 证书，也即是 CA 的成员，但无法知道对方的具体身份，从而实现双方相互匿名认证。CA 只在协议的初始化阶段招募登记成员并颁发 DAA 证书，之后不再参与协议运行，即使 CA 与 initiator 或 responder 串通，也无法破解另一方的匿名性。

(2) 用户可控关联性

在一般情况下，一个用户的两次会话是不可关联的，攻击者无法根据一个用户两次生成的 DAA 签名来判断会话是否相关联。但在特殊情况下，即 initiator 在两次会话的密钥协商中选择相同的 B，因私钥 f 不变从而得到相同的 K=Bf，那么对方可以依据相同的 B、K 判断发起两次会话的是同一个用户，但这种关联性是受 initiator 控制的。

(3) 完全前向安全

即使 initiator 和 responder 的私钥泄露，对于以前某次的会话密钥，攻击者可以根据通信记录计算 T2，T2'，但由于有限域上离散对数问题的困难性，攻击者无法从 B 和 T2 或 T2' 计算或，所以也无法计算会话密钥。

(4) 密钥支配安全

密钥协商中任何一方皆无法事先决定对方的安全参数，更无法独立控制整个密钥协商的过程，所以会话密钥是由双方共同生成的，不存在密钥支配。

(5) 已知密钥安全

任何一次会话密钥的产生都与随机值、相关，而每次会话的以及之间是不相关的，所以每次会话产生

(下转第 78 页)

(上接第 61 页)

的密钥不同，一次会话密钥的泄露不会影响其它会话密钥的安全。

5 结语

本文针对目前匿名认证密钥协商协议的主要不足——不能实现通信双方相互之间匿名和只能提供单向匿名认证，基于椭圆曲线密码学和双线性映射的 DAA 协议，在可信平台上设计了一种提供双向匿名认证的密钥协商协议，使通信双方能够相互验证对方具有某种成员关系，又不暴露各自的真实身份。此外，该协议具有密钥协商协议的一系列安全属性，提供了较好的安全性，适用于计算和存储资源有限的应用场合。

参考文献

1 Boyd C, Park D. Public Key Protocols for Wireless Communications. Proc.ICISC 1998, Springer-Verlag, 1998. 47－57.

2 Shoup V. On Formal Models for Secure Key Exchange (Version 4).Zurich: IBM Research, 1999.

3 Trusted computing group. Trusted computing platform alliance(TCPA)main specification version 1.1b, 2001. http:// www. trustedcomputinggroup.org

4 Brickell E, Camenisch J, Chen L. Direct anonymous attestation. Proc. of the 11th ACM Conference on Computer and Communications Security.ACM Press, 2004.132－145.

5 Brickell E, Chen L, Li J. Simplified Security Notions of Direct Anonymous Attestation and a Concrete Scheme from Pairings.Cryptology ePrint Archive Report 2008/104.

6 Pointcheval D, Stern J. Security proofs for signature schemes. Advances in Cryptology—EUROCRYPT'96. Springer-Verlag, 1996.387－398.

7 Ryu E, Yoon E, Yoo K. An efficient ID-based authenticated key agreement protocol from pairings. Proc.NETWORKING'04.Springer-Verlag, 2004.1458－1463.

8 Trusted Computing Group. TPM Main Specifications—Part 1 Design Principles version 1.2, 2003.