对一个强安全的认证密钥交换协议的分析

胡学先*①② 魏江宏① 叶 茂①
①(解放军信息工程大学 郑州 450002)
②(中国科学院软件研究所 北京 100190)

摘 要：在 2012 年第 15 届国际公钥密码学(PKC)年会上，Fujioka 等人利用密钥封装机制(KEM)提出了认证密钥交换(AKE)协议的一个通用构造，称为 GC 协议，并在 CK$^+$模型下证明了该协议的安全性。该文对 GC 协议进行了安全性分析，指出该协议是不安全的，难于抵抗不知道任何秘密信息的外部攻击者实施的假冒攻击，进一步分析了原协议安全性证明中被疏忽之处。

关键词：密码学；认证密钥交换；可证明安全；假冒攻击
中图分类号：TP309 **文献标识码**： A **文章编号**：1009-5896(2013)09-2278-05
DOI: 10.3724/SP.J.1146.2012.01380

Cryptanalysis of a Strongly Secure Authenticated Key Exchange Protocol

Hu Xue-Xian①② Wei Jiang-hong① Ye Mao①
①(*PLA Information Engineering University*, *Zhengzhou* 450002, *China*)
②(*Institute of Software*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*)

Abstract: In the 15th (2012) IACR international conference on practice and theory of Public-Key Cryptography (PKC), Fujioka *et al.* proposed a generic construction of Authenticated Key Exchange (AKE) from a Key Encapsulation Mechanism (KEM), which is called the GC protocol and is proven to be secure in the CK$^+$ security model. In this paper, it is pointed out by cryptanalysis that the GC protocol is not CK$^+$ secure. Concrete attacks in which the outside adversary, without knowing the static or ephemeral keys of the users, imitates a valid user are also given. Further, the errors in the original security proof are analyzed.

Key words: Cryptography; Authenticated Key Exchange (AKE); Provable security; Imitate attack

1 引言

认证密钥交换(AKE)协议是密码学中的一个基本模块，可使两个或多个参与方利用不安全的信道建立一个共享的会话密钥，供随后的密码方案使用。只有当用户之间能够生成安全的会话密钥时，上层的密码算法，如对称加密和完整性认证，才能发挥应有的功能。因此，构造安全的认证密钥交换协议对于设计安全可靠的通信系统有着基础性的重要作用[1−3]。

为了保证所设计的密钥交换协议能够抵抗各种已知攻击方法，现在普遍采用的一个方法是在严格的形式化分析模型下将协议的安全性归约到一些公认的计算困难性问题求解上。文献[4]提出了认证密钥交换的第 1 个安全性分析模型，抓住了 AKE 的基本安全需求，如已知会话密钥安全、抗伪装攻击等，但也存在敌手能力过弱、模型可移植性不强等缺点。为了提高安全模型的强度，Canetti 和 Krawczyk[5]提出了 CK 模型，考虑了部分用户的长期私钥或内部状态泄露对其它用户间的密钥协商的影响。不过，该模型没有考虑密钥泄露伪装(KCI)攻击、弱前向安全(wPFS)和最大暴漏(MEX)攻击。2005 年 Krawczy 在 CK 模型基础上进一步提出了 CK$^+$模型[6]，除了 CK 模型的所有安全目标之外，还涵盖了相对于 KCI 攻击、wPFS 和 MEX 攻击的安全性。

利用提出的 CK$^+$模型，Krawczy 在著名的 MQV 协议基础上设计了一个安全性有较大提升的 HMQV 协议[6]。不过，该协议的安全性证明需要用到随机预言机模型(ROM)[7]。而 Dent[8]指出，在 ROM 下证明安全的协议在实际中可能会存在安全性缺陷。Boyd 等人[9]基于密钥封装机制(KEM)提出了一个标准模型下安全的密钥交换协议，在 CK 模型中可证明安全且能抵抗 KCI 攻击。然而，该协议并不能抵抗 MEX 攻击，因此并不是 CK$^+$安全的。为了设计具有更强安全性并且不依赖于 ROM 的 AKE 协议，日本学者 Fujioka 等人[10]在 2012 年 PKC 年会上提出了 AKE 协议的一个通用构造，称为 GC 协议，并在 CK$^+$模型下证明了该协议的安全性。

本文对 Fujioka 等人[10]提出的 GC 协议进行了分

2012-10-26 收到，2013-05-30 改回
国家 973 计划项目(2012CB315905)和河南省科技攻关计划项目(122102210225)资助课题
*通信作者：胡学先 xuexian_hu@yahoo.com.cn

析，指出该协议并不是 CK$^+$安全的，给出了一个拥有 CK$^+$模型中所描述能力的外部攻击者实施的假冒协议发起方(或响应方)的具体攻击方法。由于原文在 CK$^+$安全性分析模型中证明了协议的安全性，还进一步分析了原协议安全性证明中的错误。

2　Fujioka 等人提出的 GC 协议

首先介绍 Fujioka 等人在 2012 年 PKC 年会上提出的 GC 协议。设 k 是安全参数，(KeyGen, EncCap, DeCap) 是一个 CCA 安全的密钥封装机制，(wKeyGen, wEnCap,wDeCap)是一个 CPA 安全的密钥封装机制。$F, F' : \{0,1\}^* \times \mathcal{FS} \to \mathcal{RS}_E$ 和 $G : \{0,1\}^* \times \mathcal{FS} \to \{0,1\}^k$ 是伪随机函数簇，其中 $\mathcal{FS}$ 是所对应的密钥空间，$\mathcal{RS}_E$ 是 KEM 中封装算法 EnCap 和 wEnCap 的随机数空间。$\mathrm{Ext} : \mathcal{SS} \times \mathcal{KS} \to \mathcal{FS}$ 是一个强随机性提取器，其中 $\mathcal{SS}$ 是提取器的种子空间，$\mathcal{KS}$ 是 KEM 的密钥空间。假设在系统初始化阶段，每个用户 U_I 均匀随机地选择参数 $\sigma_I \in \mathcal{FS}$，$r_I \in \mathcal{RS}_G$，并运行 KEM 中的密钥生成算法 $\mathrm{KeyGen}(1^k, r_I)$ 得到密钥对 $(ek_{I,1}, dk_{I,1})$，其中 $\mathcal{RS}_G$ 是 KEM 中密钥生成算法的随机数空间。此时，用户 U_I 的私钥是 $(dk_{I,1}, \sigma_I)$，公钥为 $ek_{I,1}$。

设拥有密钥对 $((dk_{A,1}, \sigma_A), ek_{A,1})$ 的用户 U_A 准备作为发起方，与拥有密钥对 $((dk_{B,1}, \sigma_B), ek_{B,1})$ 的响应方 U_B 进行密钥交换，GC 协议的具体步骤如下：

(1)用户 U_A 随机地选择临时私钥 $r_{A,1}, r'_{A,1} \in \mathcal{FS}$, $r_{A,2} \in \mathcal{RS}_G$，利用密钥封装算法 EnCap 生成 $(CT_{A,1}, K_{A,1}) \leftarrow \mathrm{EnCap}_{ek_{B,1}}(F_{\sigma_A}(r_{A,1}) \oplus F'_{r'_{A,1}}(\sigma_A))$，运用密钥生成算法 wKeyGen 得到临时密钥对 $(ek_{A,2}, dk_{A,2}) \leftarrow w\mathrm{KeyGen}(1^k, r_{A,2})$，然后发送消息 $(U_A, U_B, CT_{A,1}, ek_{A,2})$ 给用户 U_B；

(2)用户 U_B 收到消息 $(U_A, U_B, CT_{A,1}, ek_{A,2})$ 后，随机地选择临时密钥 $r_{B,1}, r'_{B,1} \in \mathcal{FS}$，$r_{B,2} \in \mathcal{RS}_E$，计算 $(CT_{B,1}, K_{B,1}) \leftarrow \mathrm{EnCap}_{ek_{A,1}}(F_{\sigma_B}(r_{B,1}) \oplus F_{r'_{B,1}}(\sigma_B))$ 和 $(CT_{B,2}, K_{B,2}) \leftarrow w\mathrm{EnCap}_{ek_{A,2}}(r_{B,2})$，并发送消息 $(U_A, U_B, CT_{B,1}, CT_{B,2})$ 给用户 U_A；用户 U_B 还依据所接收到的消息解封装得到密钥 $K_{A,1} \leftarrow \mathrm{DeCap}_{dk_{B,1}}(CT_{A,1})$，利用随机性提取器计算 $K'_1 = \mathrm{Ext}(s, K_{A,1})$, $K'_2 = \mathrm{Ext}(s, K_{B,1})$, $K'_3 = \mathrm{Ext}(s, K_{B,2})$, 设定会话记录为 $ST = (U_A, U_B, ek_{A,1}, ek_{B,1}, CT_{A,1}, ek_{A,2}, CT_{B,1}, CT_{B,2})$，基于伪随机函数簇计算得到与用户 U_A 共享的会话密钥 $SK = G_{K'_1}(ST) \oplus G_{K'_2}(ST) \oplus G_{K'_3}(ST)$。最后，用户 U_B 标识该会话为已完成并擦除所有会话状态；

(3)用户 U_A 收到消息 $(U_A, U_B, CT_{B,1}, CT_{B,2})$ 后，利用解封装密钥 $dk_{A,1}, dk_{A,2}$ 解封装得到密钥 $k_{B,1} \leftarrow \mathrm{DeCap}_{dk_{A,1}}(CT_{B,1})$，$k_{B,2} \leftarrow w\mathrm{DeCap}_{dk_{A,2}}(CT_{B,2})$，进一步计算得到 $K'_1 = \mathrm{Ext}(s, K_{A,1})$，$K'_2 = \mathrm{Ext}(s, K_{B,1})$，$K'_3 = \mathrm{Ext}(s, K_{B,2})$，设定会话记录为 $ST = (U_A, U_B, ek_{A,1}, ek_{B,1}, CT_{A,1}, ek_{A,2}, CT_{B,1}, CT_{B,2})$，计算与用户 U_B 共享的会话密钥 $SK = G_{K'_1}(ST) \oplus G_{K'_2}(ST) \oplus G_{K'_3}(ST)$。最后，标识该会话为已完成并擦除所有会话状态。

特别地，GC 协议规定用户 U_A 的会话状态(session state)包含：临时私钥 $(r_{A,1}, r'_{A,1}, r_{A,2})$，KEM 的密钥 $(K_{A,1}, K_{B,1}, K_{B,2})$，随机性提取器的输出 (K'_1, K'_2, K'_3)，以及所涉及到的伪随机函数簇的输出 $(F_{\sigma_A}(r_{A,1}), F'_{r'_{A,1}}(\sigma_A), G_{K'_1}(ST), G_{K'_2}(ST), G_{K'_3}(ST))$。用户 U_B 的会话状态按照类似的方式予以定义。

3　对 GC 协议的攻击

Fujioka 等人认为，GC 协议的提出解决了“在目前最强的安全模型中设计基于一般假设、不依赖于 ROM 的安全 AKE 协议”的公开问题[10]。具体地说，他们认为 GC 协议除了达到基本的语义安全和弱前向安全(wPFS)外，还能够抵抗密钥泄露伪装(KCI)攻击和最大暴漏(MEX)攻击。其中，KCI 攻击是指，攻击者在已知某些用户被泄露的私钥的情况下，伪装成诚实的用户参与协议运行，达到欺骗泄露私钥的拥有方的目的；MEX 攻击是指，攻击者在知晓测试会话中参与方的某些长期和临时私钥的情况下，试图以不可忽略的概率区分会话密钥和相同长度的均匀随机值。为了支持他们的结论，Fujioka 等还在 Krawczyk 提出的 CK$^+$模型中“证明了”GC 协议的安全性，即对具有模型中描述能力的任意攻击者，其区分一个测试会话返回的值是真实密钥还是随机值的优势是可忽略的。

本文指出，GC 协议并不是 CK$^+$安全的。一个具备 CK$^+$模型中所描述能力的外部攻击者，在无需知道任何合法用户私钥的条件下，就可以冒充合法的用户作为 GC 协议的发起方(或响应方)与其他用户进行密钥交换，并得到最终的会话密钥。

3.1 CK$^+$安全性分析模型

本节首先简要介绍 Krawczy 改进的包含了 KCI 攻击的 CK$^+$模型[6,10]。

模型假设共有 n 个用户参与通信，每个用户 U_i 中可能同时运行协议的多个会话(session)，被模型化为概率多项式时间图灵机。每个会话都拥有唯一的会话标识 sid，定义为会话发送和接受的所有消息的级联。会话在被激活后，将按照协议规范运行并产生相应的输出消息或者是会话密钥，称生成了会话密钥的会话是完成的。如果两个会话的拥有方互为对方的意定通信方，并且会话标识中包含的发送和接受的消息相互匹配，就称这两个会话为匹配会话。

进一步还假设有一个被模型化为概率多项式时间

图灵机的攻击者 $\mathcal{A}$。攻击者 $\mathcal{A}$ 控制着网络的所有通信，能够随意修改、丢弃和伪造消息，能够编排消息的发送顺序或者修改消息的接受方。为了描述攻击者得到部分私密信息的条件下会话密钥的安全性，还允许攻击者 $\mathcal{A}$ 发出下列询问：

(1) Send(U_i, message)：模型化针对用户的主动攻击，攻击者向用户发送任何消息，询问输出会话在收到消息后的响应消息；

(2) SessionKeyReveal(sid)：模型化会话密钥的泄露，攻击者通过该询问可以得到已经完成的会话 sid 的会话密钥；

(3) SessionStateReveal(sid)：模型化会话内部状态信息的泄露。如果会话 sid 没有完成，则提供给攻击者该会话的会话状态(session state)。会话状态所包含的信息根据不同协议进行不同的定义，但一般包含临时私钥和所有不及时擦除的中间状态信息，不包含用户的长期私钥；

(4) Corrupt(U_i)：通过这个询问，攻击者可以得到用户 U_i 的所有内部信息。如果攻击者针对用户 U_i 进行了该询问，就称这个用户是不诚实的；否则，称该用户是诚实的。

称一个会话 sid^* 是新鲜的(fresh)，如果拥有该会话的用户是诚实的，且(1)在 sid^* 的匹配会话不存在的情况下，没有对会话 sid^* 进行 SessionKeyReveal 和 SessionStateReveal 询问；(2)若存在 sid^* 的匹配会话 $\overline{\text{sid}}^*$，没有对会话 sid^* 和 $\overline{\text{sid}}^*$ 中的任何一个进行 SessionKeyReveal 和 SessionStateReveal 询问。

为了定义安全性，模型考虑攻击者 $\mathcal{A}$ 和协议用户会话之间进行下述安全游戏。攻击者可以随机激活会话，并按照任意顺序发送上述 4 种询问。在其中某个时刻，攻击者可以对某个新鲜的、已经完成的会话发出下述测试询问：

Test(sid^*)：选择一个均匀随机的比特 $b\in\{0,1\}$，如果 $b=0$，返回给攻击者会话 sid^* 所拥有的真实的会话密钥；如果 $b=1$，返回给攻击者一个均匀分布的随机值。

攻击者在测试询问之后可以继续上述 4 种询问，不过不能违反测试会话的安全性定义。在游戏最后，攻击者 $\mathcal{A}$ 输出一个比特 b' 作为对随机比特 b 的猜测。如果 $b'=b$，认为攻击者猜测成功，记这个事件为 $\text{Succ}^{\mathcal{A}}$。在被分析的协议为 Π 时，定义攻击者 $\mathcal{A}$ 的优势为 $\text{Adv}_{\Pi}^{\text{AKE}}(\mathcal{A})=2\text{Pr}[\text{Succ}^{\mathcal{A}}]-1$。

定义 1 (CK$^+$安全性) 称一个密钥交换协议 Π 是安全的，如果下述两个条件成立：

(1)若两个未被腐化的用户完成了匹配会话，则它们生成的会话密钥相同；

(2)若攻击者 $\mathcal{A}$ 没有同时得到测试会话 sid^* 的长期私钥和临时私钥，也没有同时得到其匹配会话 $\overline{\text{sid}}^*$ (如果 $\overline{\text{sid}}^*$ 存在的情况下)的长期私钥和临时私钥，则其攻击优势是安全参数的可忽略量。

3.2 假冒协议发起方的攻击

设用户 U_I 的私钥是 $(dk_{I,1},\sigma_I)$，公钥为 $ek_{I,1}$，攻击者 E 在不知道诚实用户 U_A,U_B 的长期和临时私钥的情况下，冒充用户 U_A 作为发起方与用户 U_B 进行密钥协商的具体步骤如下：

(1)攻击者 E 随机地选择临时私钥 $r_1\in\mathcal{RS}_E$，$r_2\in\mathcal{RS}_G$，利用密钥封装算法 EnCap 和 U_B 的公钥 $ek_{B,1}$ 计算得到 $(CT_{A,1},K_{A,1})\leftarrow\text{EnCap}_{ek_{B,1}}(r_1)$，利用密钥生成算法 wKeyGen 得到临时密钥对 $(ek_{A,2},dk_{A,2})\leftarrow w\text{KeyGen}(1^{\mathcal{K}},r_2)$，然后冒充用户 U_A 发送消息 $(U_A,U_B,CT_{A,1},ek_{A,2})$ 给用户 U_B；

(2)用户 U_B 收到消息 $(U_A,U_B,CT_{A,1},ek_{A,2})$ 后，认为 U_A 希望作为发起方与其进行密钥协商，故其按照 GC 协议规范作为响应方予以回应：随机地选择临时密钥 $r_{B,1},r'_{B,1}\in\mathcal{FS}$，$r_{B,2}\in\mathcal{RS}_E$，计算 $(CT_{B,1},K_{B,1})\leftarrow\text{EnCap}_{ek_{A,1}}(F_{\sigma_B}(r_{B,1})\oplus F_{r'_{B,1}}(\sigma_B))$ 和 $(CT_{B,2},K_{B,2})\leftarrow w\text{EnCap}_{ek_{A,2}}(r_{B,2})$，并发送消息 $(U_A,U_B,CT_{B,1},CT_{B,2})$ 给用户 U_A；用户 U_B 还依据所接收到的消息解封装得到密钥 $K_{A,1}\leftarrow\text{DeCap}_{dk_{B,1}}(CT_{A,1})$，利用随机性提取器计算 $K'_1=\text{Ext}(s,K_{A,1}),K'_2=\text{Ext}(s,K_{B,1}),K'_3=\text{Ext}(s,K_{B,2})$，设定会话记录为 $ST=(U_A,U_B,ek_{A,1},ek_{B,1},CT_{A,1},ek_{A,2},CT_{B,1},CT_{B,2})$，基于伪随机函数簇计算得到与用户 U_A 共享的会话密钥 $SK=G_{K'_1}(ST)\oplus G_{K'_2}(ST)\oplus G_{K'_3}(ST)$。最后，用户 U_B 标识该会话为已完成并擦除所有会话状态；

(3)攻击者 E 截获用户 U_B 发送给用户 U_A 的消息 $(U_A,U_B,CT_{B,1},CT_{B,2})$，保存 $CT_{B,1}$，利用私钥 $dk_{A,2}$ 解封装 $CT_{B,2}$ 得到 $K_{B,2}\leftarrow w\text{DeCap}_{dk_{A,2}}(CT_{B,2})$；

(4)攻击者 E 激活用户 U_A 中的一个新的会话，让 U_A 作为发起方与某个诚实用户 U_C 进行密钥协商。其中，用户 U_A 按照协议规范计算并发送第 1 条消息 $(U_A,U_C,\widetilde{CT}_{A,1},\widetilde{ek}_{A,2})$ 给用户 U_C，用户 U_C 收到上述消息后计算并发送响应消息 $(U_A,U_C,CT_{C,1},CT_{C,2})$ 给用户 U_A。此时，攻击者 E 截获消息 $(U_A,U_C,CT_{C,1},CT_{C,2})$，将其修改为 $(U_A,U_C,CT_{B,1},CT_{C,2})$ 后假冒用户 U_C 的身份发送给用户 U_A；

(5)用户 U_A 收到消息 $(U_A,U_C,CT_{B,1},CT_{C,2})$ 后，将利用其解封装算法 DeCap 和私钥 $dk_{A,1}$ 对密文 $CT_{B,1}$ 解封装得到 $K_{B,1}\leftarrow\text{DeCap}_{dk_{A,1}}(CT_{B,1})$。此时，$E$ 对 U_A 中与消息 $(U_A,U_C,CT_{B,1},CT_{C,2})$ 相关联的会话进行 SessionStateReval 询问，得到该会话的所有会话状态，

从而得到 $K_{B,1}$；

(6)攻击者已知晓 $K_{A,1},K_{B,1},K_{B,2}$，利用随机性提取器计算 $K_1^{'}=\mathrm{Ext}(s,K_{A,1}),K_2^{'}=\mathrm{Ext}(s,K_{B,1}),K_3^{'}=\mathrm{Ext}(s,K_{B,2})$，生成 ST=$(U_A,U_B,ek_{A,1},ek_{B,1},CT_{A,1},ek_{A,2},CT_{B,1},CT_{B,2})$，计算得到会话密钥 $SK=G_{K_1^{'}}(ST)\oplus G_{K_2^{'}}(ST)\oplus G_{K_3^{'}}(ST)$。

3.3 假冒协议响应方的攻击

尽管 GC 协议对于发起方和响应方是一个非对称的协议，攻击者 E 仍能以类似的方式假冒协议响应方 U_B 与诚实用户 U_A 进行密钥交换，具体步骤如下：

(1)攻击者 E 激活用户 U_A，让 U_A 作为发起方与用户 U_B 进行密钥交换。此时，用户 U_A 选择随机值 $r_{A,1},r_{A,1}^{'}\in\mathcal{FS},r_{A,2}\in\mathcal{RS}_G$，计算 $(CT_{A,1},K_{A,1})\leftarrow\mathrm{EnCap}_{ek_{B,1}}(F_{\sigma_A}(r_{A,1})\oplus F_{r_{A,1}^{'}}^{'}(\sigma_A))$，运用密钥生成算法 wKeyGen 得到临时密钥对

$$(ek_{A,2},dk_{A,2})\leftarrow w\mathrm{KeyGen}\quad(1^k,r_{A,2})$$

然后发送消息 $(U_A,U_B,\ CT_{A,1},ek_{A,2})$ 给用户 U_B；

(2)攻击者 E 截获用户 U_A 发送给用户 U_B 的消息 $(U_A,U_B,CT_{A,1},ek_{A,2})$。然后，攻击者 E 选择随机值 $r_1,r_2\in\mathcal{RS}_E$，计算 $(CT_{B,1},K_{B,1})\leftarrow\mathrm{EnCap}_{ek_{A,1}}(r_1)$ 和 $(CT_{B,2},K_{B,2})\leftarrow w\mathrm{EnCap}_{ek_{A,2}}(r_2)$，并发送消息 $(U_A,U_B,CT_{B,1},CT_{B,2})$ 给用户 U_A；

(3)攻击者 E 激活用户 U_C，让其作为发起方与用户 U_B 发起一个新的会话。在用户 U_C 按照协议规范计算并发送消息 $(U_C,U_B,CT_{C,1},ek_{C,2})$ 给用户 U_B 时，攻击者将该消息截获、修改成为 $(U_C,U_B,CT_{A,1},ek_{C,2})$ 再发送给用户 U_B；

(4)用户 U_B 收到消息 $(U_C,U_B,CT_{A,1},ek_{C,2})$ 后，将利用其私钥 $dk_{A,1}$ 对所收到的消息中的 $CT_{A,1}$ 进行解封装得到 $K_{A,1}\leftarrow\mathrm{DeCap}_{dk_{B,1}}(CT_{A,1})$。此时，攻击者 E 对 U_B 中与 U_C 相关联的会话进行 SessionStateReval 询问，得到封装密钥 $K_{A,1}$；

(5)攻击者 E 已知晓封装密钥 $K_{A,1},K_{B,1},K_{B,2}$，并知道用户 U_A 发送和收到的所有消息，即会话脚本 $ST=(U_A,U_B,ek_{A,1},ek_{B,1},CT_{A,1},ek_{A,2},CT_{B,1},CT_{B,2})$，因此可以直接计算得到 U_A 拥有的会话密钥 $SK=G_{K_1^{'}}(ST)\oplus G_{K_2^{'}}(ST)\oplus G_{K_3^{'}}(ST)$。

上述攻击中，攻击者无需知道任何诚实用户的长期和临时私钥，仅需具备截获、修改用户间的通信消息、进行 SessionStateReval 询问的能力，就可以成功地实施假冒攻击并得到最终的会话密钥。注意到两个攻击中 SessionStateReval 询问的对象既不是测试会话，也不是测试会话的匹配会话，因此进行该询问并未违反 CK$^+$模型中关于测试会话的新鲜性定义，这表明 GC 协议不是 CK$^+$安全的。进一步，由于 CK 模型[5]中也允许攻击者进行 SessionStateReval 询问，从而可知 GC 协议实际上甚至不是 CK 安全的。

4 分析与改进

4.1 对原协议安全性证明错误之处的分析

Fujioka 等在 Krawczyk 提出的 CK$^+$模型[6]中给出了 GC 协议的安全性证明[10]，因此如果证明中不存在错误的情况下，协议应该到达 CK$^+$模型定义的安全性目标。上述假冒攻击表明原协议的安全性证明中必定存在错误或疏忽，本节将分析证明过程中疏忽的细节，以求在以后的安全 AKE 协议的设计和分析过程中能够有所借鉴。

文献[10]采用了一系列逐步修改的混合实验将 CK$^+$模型所对应的真实实验和测试会话中的密钥被替换成随机值的实验联系起来，并通过考察每两个相邻的混合实验之间攻击者成功优势差达到限定真实的协议中攻击者成功优势的目的。具体地，原证明根据测试会话是否存在匹配会话，参与方的哪些秘密信息被泄露等不同情况考虑了 $E_1,E_2,\cdots,E_8$ 共 8 个事件，并利用混合实验序列对每个事件发生条件下攻击者的成功概率进行估计。例如，为考虑事件 E_1(测试会话 sid^* 无匹配会话，其拥有方是会话的发起方，其长期密钥已经泄露)发生的条件下，攻击者的成功概率，共采用了 7 个混合实验 $H_0,H_1,\cdots,H_6$，逐步将 sid^* 中的会话密钥替换成均匀选取的随机值。其中混合游戏 H_3 将测试会话中的关于 $(CT_{A,1}^*,K_{A,1}^*)$ 的计算方式由 $(CT_{A,1}^*,K_{A,1}^*)\leftarrow\mathrm{EnCap}_{ek_{B,1}}(F_{\sigma_A}(r_{A,1})\oplus F_{r_{A,1}^{'}}^{'}(\sigma_A))$ 修改成 $(CT_{A,1}^*,K_{A,1}^*)\leftarrow\mathrm{EnCap}_{ek_{B,1}}(F_{\sigma_A}(r_{A,1})\oplus RF(\sigma_A))$，混合游戏 H_4 进一步将 $K_{A,1}^*$ 替换成从密钥空间 $\mathcal{KS}$ 中完全随机选取的值 $K_{A,1}^*\leftarrow\mathcal{KS}$，并“证明了”若存在协议攻击者能以不可忽略的概率区分实验 H_3 和 H_4，则可以构造针对所用 IND-CCA 安全性的 KEM 攻击者 $\mathcal{S}$ 以不可忽略的优势成功。

攻击者能够成功地假冒协议响应方与用户 U_A 进行密钥协商并得到最终的会话密钥表明 KEM 攻击者 $\mathcal{S}$ 的构造过程存在错误。在原证明中，KEM 攻击者 $\mathcal{S}$ 收到公开密钥 ek^* 后，为除用户 U_B 外的所有用户选择长期密钥，并将 U_B 的公钥设定为 ek^*，然后模拟协议运行与协议攻击者进行交互，当协议攻击者进行 CK$^+$模型中所规定的谕示询问时，$\mathcal{S}$ 进行模拟回答。对于询问 SessionStateReval(sid)，攻击者返回相应的临时私钥和所有的中间计算结果，若 sid 的拥有方是 U_B，$\mathcal{S}$ 将 U_B 收到的密文提交给相应的解密谕示，然后就可以模拟所有的中间计算结果。我们指出这样的模拟是不正确的。类似于 3.2 节的步骤(3)，若协议攻击者先用消息 $(U_C,U_B,CT_{A,1}^*,ek_{C,2})$ 激活用户 U_B 中的一个新的

会话，再进行SessionStateReval(sid)询问，依据CCA安全性游戏规定，$\mathcal{S}$并不能将U_B收到的密文$CT_{A,1}^{*}$提交给相应的解密谕示。因此，$\mathcal{S}$不能完全地模拟协议运行，其构造的模拟环境既不同于实验H_3也不同于实验H_4。这说明，即使协议所采用的密钥封装机制(KeyGen,EnCap,DeCap)是CCA安全的，即不存在概率多项式时间KEM攻击者能够以不可忽略的优势成功，也不能保证协议攻击者不能区分实验H_3和H_4，这是原证明中主要被疏忽之处。

4.2 改进措施

从上述攻击和安全性分析可以看出，攻击者能够成功的主要原因是每个用户的会话状态包含了过多的信息。尽管Fujioka等提出的协议[10]采用纠缠伪随机函数技术抵抗了临时私钥泄露，使得攻击者不能够利用得到的会话临时私钥或长期私钥计算出$K_{A,1},K_{B,1}$，但是攻击者总能够利用其它的会话作为解密预言机得到这些信息。相比而言，Boyd等的原始协议[9]反而不存在这类问题。这是由于Boyd等注意到一个会话在收到密钥封装消息后，解封装得到的密钥将会立即被用于计算会话密钥，故可以认为它们存在的时间极短，从而不属于会话状态。因此可以看出，一个直观的改进方法就是(类似于文献[9])要求用户A在收到封装密文$CT_{B,1}$后，将解密得到的密钥$K_{B,1}$立即用于计算会话密钥并擦除，即显式地要求$K_{B,1},K_2^{'}$不属于用户A的会话状态。类似地，也显式地要求$K_{A,1},K_1^{'}$不属于用户B的会话状态。

5 结束语

本文对Fujioka等最近在2012年PKC年会上提出的GC协议的分析表明该协议是不安全的，难于抵抗不知道任何秘密信息的外部攻击者实施的假冒攻击，进一步分析了原协议安全性证明中的错误。协议不安全的主要原因是协议设计者错误地将过多的信息放在会话状态中，并允许攻击者进行询问，使得攻击者可以将诚实的会话实例当作KEM解封装谕示使用。本文攻击也再次表明了，尽管可证明安全理论是协议分析设计的主要研究方法之一，协议的设计和安全性证明是技巧性强且易于出错部分，需要谨慎对待。

参 考 文 献

[1] Smart N. Update to provable security: design and open questions[OL]. Technical Report D.AZTEC.5, 2007. http://www.ecrypt.org/documents.2008.9.

[2] Pointcheval D. Password-based authenticated key exchange[C]. Proceedings of 15th IACR International Conference on Practice and Theory of Public-Key Cryptography (PKC 2012), Darmstadt, Germany, May 21–23, 2012: 390–397.

[3] Guo Y and Zhang Z F. Authenticated key exchange with entities from different settings and varied groups[C]. Proceedings of 6th International Conference on Provable Security (ProvSec 2012), Chengdu, China, September 26–28, 2012: 276–287.

[4] Bellare M and Rogaway P. Entity authentication and key distribution[C]. Proceedings of CRYPTO 1993, California, USA, August 22–26, 1993: 232–249.

[5] Canetti R and Krawczyk H. Analysis of key-exchange protocols and their use for building secure channels[C]. Proceedings of EUROCRYPT 2001, Innsbruck, Austria, May 6–10, 2001: 453–474.

[6] Krawczyk H. HMQV: a high-performance secure Diffie-Hellman protocol[C]. Proceedings of CRYPTO 2005, California, USA, August 14–18, 2005: 546–566.

[7] Canetti R, Goldreich O, and Halevi S. The random oracles methodology, revisited[C]. Proceedings of the 30th Annual ACM Symposium on the Theory of Computing (STOC 1998), Dallas, Texas, USA, May 23–26, 1998: 209–218.

[8] Dent A W. Adapting the weaknesses of the random oracle model to the generic group model[C]. Proceedings of ASIACRYPT 2002, Queenstown, New Zealand, December 1–5, 2002: 100–109.

[9] Boyd C, Cliff Y, Nieto J G, *et al*.. Efficient one-round key exchange in the standard model[C]. Proceedings of ACISP 2008, Wollongong, Australia, July 7–9, 2008: 69–83.

[10] Fujioka A, Suzuki K, Xagawa K, *et al*.. Strongly secure authenticated key exchange from factoring, codes, and lattices[C]. Proceedings of 15th (2012) IACR International Conference on Practice and Theory of Public-Key Cryptography (PKC), Germany, May 21–23, 2012: 467–484.

胡学先： 男，1982年生，博士，讲师，研究方向为安全协议.

魏江宏： 男，1987年生，博士生，研究方向为安全协议.

叶 茂： 男，1988年生，硕士生，研究方向为安全协议.