doi:10.3969/j.issn.1000-436x.2014.03.012

用于云存储的安全容错编码

谭鹏许[1]，陈越[1]，兰巨龙[2]，贾洪勇[1]
（1. 解放军信息工程大学 网络空间安全学院，河南 郑州 450004；2. 国家数字交换系统工程技术研究中心，河南 郑州 450002）

摘 要：针对当前基于 RC 编码的容错技术的安全缺陷，提出了一种安全编码——SRCS 编码，以保证在云计算以及云存储这种高度开放环境下，存储系统容错过程中数据的安全性。该编码将门限体制引入到了传统的 RC 编码当中，利用基于公钥的门限体制保护编码矩阵，在确保基于传统 RC 编码的容错技术高效、低冗余优势的前提下，解决了其在开放环境下编码矩阵存在的安全问题。最后利用判定性 BDHE 假设，在部分适应性攻击模型下证明了 SRCS 编码的安全性。

关键词：RC 编码；容错技术；云计算；云存储；基于公钥的门限体制；判定性 BDHE 假设
中图分类号：TP309.3 **文献标识码**：A **文章编号**：1000-436X(2014)03-0109-07

Secure fault-tolerant code for cloud storage

TAN Peng-xu[1], CHEN Yue[1], LAN Ju-long[2], JIA Hong-yong[1]
(1. Institute of Cyberspace Security PLA Information Engineering University, Zhengzhou 450004, China;
2. National Digital Switching System Engineering & Technological Research Center, Zhengzhou 450002, China)

Abstract: A secure regenerating code, called SRCS was proposed to solve the security problem during the process of the fault-tolerant of storage systems, especially the storage system in an extremely open environment such as cloud computing and cloud storage. SRCS achieves the security of the encoding matrix in the regenerating code using the threshold public-key encryption with low redundancy and high efficiency. It is proven that SRCS is secure in the semi-adaptive model using decisional BDHE assumption.

Key words: regenerating codes; fault-tolerant; cloud computing; cloud storage; threshold public-key encryption; decisional BDHE assumption

1 引言

伴随着云计算的高速发展，云存储技术得到了广泛的关注。云存储（cloud storage）这个概念一经提出，就得到了众多厂商的支持和关注。Amazon 依据自身的云服务平台推出了 Elastic Compute Cloud（EC2，弹性计算云）云存储产品，旨在为用户提供互联网服务形式同时提供更强的存储和计算功能。内容分发网络服务提供商 CDNetworks 和业界著名的云存储平台服务商 Nirvanix 发布了一项新的合作，并宣布结成战略伙伴关系，以提供业界目前唯一的云存储和内容传送服务集成平台。除此之外，Google 推出了 GFS（Google file system），EMC 推出了 Atoms。作为一种服务，云存储需要向用户提供可靠、高效的大规模数据服务，并且有服务运营商保证数据的安全性以及可用性。但是，IBM 公司的一份报告[1]指出，在 IBM 公司的云数据中心内，每个 Map-Reduce 过程中至少有 5 个存储节点意外失效；在一个拥有 4 000 个节点的数据服务站中，平均每 6 h 就会有一块存储硬盘失效。除了 IBM 公司外，Amazon 的 S3 系统以及 Google 的 Google Docs 都出现过由于存储节点失效而造成的系统崩溃事件，给数以千计的企业用户和个人用户造成了不可挽回的损失。由此可以看出，云存储系统的可靠性是制约其发展的关键，而数据容错技术则是实现云存储系统可靠性的关键。

收稿日期：2012-12-27；**修回日期**：2013-03-04
基金项目：国家科技支撑计划基金资助项目(2008BAH37B03)
Foundation Item: National Key Technology R&D Program of China (2008BAH37B03)

与大多数的分布式存储系统相同，当前云存储中也存在有 2 种类型的容错：基于副本的容错技术和基于存储冗余的容错技术。后者中最具代表性的技术便是基于纠删码（erasure code）的容错技术。在文献[2]中已经论证了在存储容量和存储代价方面，基于纠删码的容错技术比基于副本的容错技术更加具备优势。然而其问题却在于基于纠删码的容错技术在容错修复时需要较大的带宽消耗。Dimakis 领导的研究小组基于纠删码提出了 RC(regenerating code)[3~5]，与纠删码相比 RC 极大地降低了修复时所需的网络带宽。Hu[6,7]，Shum[8,9] 以及 Kermarrec[10] 优化了 RC 的构造模型进一步降低了网络带宽消耗。Li 等人[11]则优化了 RC 的修复模式，在保证修复效率的前提下降低了网络带宽消耗。

上述编码都单纯地从容错修复的角度出发来进行设计，而没有考虑到容错过程中的安全问题。然而，在目前复杂的网络环境中，充斥着网络攻击，整个系统在容错过程中的安全性也是不得不考虑的问题。目前已经有学者开始着手这方面的研究，Gkantsidis[12]提出了一种针对 P2P 模式文件存储系统的安全编码，它将可能出现的数据篡改问题考虑到容错过程当中，提出了数据节点之间的相互认证以保证数据的安全性；Li[13]通过实验分析讨论了如何通过网络编码实现分布式存储中数据的安全性以及存储效率的问题；Lin[14]在 Dimakis 前期研究[15]的基础上提出了一种安全的 MDS 编码，利用了门限加密的思想，只有一定数量的密钥服务器同时保证接收者的安全性，数据才得以恢复。Gkantsidis 的方法由于在构建安全框架时依赖于 P2P 模式的网络结构，因而存在应用场景的局限；Li 只是对其所设想的方案进行了理论分析和实验，并没有给出切实的实现方案；Lin 设计的方案则由于编码本身的问题和其所采用的将多个文件进行混合编码的方式，存在有一定的概率不能对存储数据进行解码，而且其在构建门限加密算法时，其安全模型较为简单，这使其只能够抵抗选择明文攻击，安全性较弱[16]。

本文旨在提出一种基于 RC 思想的、高安全性的、可用于云存储系统的容错编码存储技术：部分适应性安全 RC（SRCS, secure regenerating code with semi-adaptive）编码。本文所提的编码存储技术利用了门限体制，只有同时获取大于等于 t 个子密钥的存储节点才能够参与到系统的容错服务中，极大地降低了利用非法节点，通过串谋获取数据，对存储数据进行污染进而攻击系统的可能性，而且 SRCS 可以抵御部分适应性攻击，具备了较强的安全性；编码构造的理论基础是基于 RC 的，继承了 RC 存储冗余小、节点恢复效率高、节点恢复所需网络总带宽较小的特点。

2 预备知识

2.1 RC

RC 采用了网络信息流图模型对存储系统修复进行建模。其假设源将数据编码存储到 n 个存储节点当中，当拥有大于等于 k 个节点可用时，即可对数据进行恢复。当节点失效时，系统会从 d（$d \geqslant k$）个节点中获取数据用以重建节点。

依据文献[17]中的假设，令消息矩阵 $\boldsymbol{M}$ 为 $d \times d$ 对称矩阵，编码矩阵 $\boldsymbol{\Psi}$ 为 $n \times d$ 矩阵，表示为：$\boldsymbol{\Psi} = [\boldsymbol{\Phi} \quad \boldsymbol{\Delta}]$，其中 $\boldsymbol{\Phi}$ 和 $\boldsymbol{\Delta}$ 分别为 $n \times k$ 和 $n \times (d-k)$ 矩阵，且满足 $\boldsymbol{\Psi}$ 中的任意 d 行是线性无关的，$\boldsymbol{\Phi}$ 中的任意 k 行是线性无关的。那么，编码后的存储矩阵为 $n \times d$ 矩阵 $\boldsymbol{C} = \boldsymbol{\Psi} \times \boldsymbol{M}$。令 $\boldsymbol{\psi}_f^{\mathrm{t}}$（$\boldsymbol{\psi}_f^{\mathrm{t}}$ 为 $\boldsymbol{\psi}_f$ 的转置）表示失效节点 f 在编码矩阵 $\boldsymbol{\Psi}$ 对应的行，那么其所存储的 d 个元素可以为向量 $\boldsymbol{\psi}_f^{\mathrm{t}}\boldsymbol{M}$。被恢复的新节点将随机连接 d 个可用节点以获取数据，这 d 个节点被表示为 $\{h_j \mid j = 1, \cdots, d\}$，且每个节点向新节点传输的数据为 $\boldsymbol{\psi}_{h_i}^{\mathrm{t}}\boldsymbol{M}\boldsymbol{\psi}_f$。因而，新节点能够获取 d 个元素，这些元素可以被表示为 $\boldsymbol{\Psi}_{\mathrm{repair}}\boldsymbol{M}\boldsymbol{\psi}_f$，其中，

$$\boldsymbol{\Psi}_{\mathrm{repair}} = \begin{bmatrix} \boldsymbol{\psi}_{h_1}^{\mathrm{t}} \\ \boldsymbol{\psi}_{h_2}^{\mathrm{t}} \\ \vdots \\ \boldsymbol{\psi}_{h_d}^{\mathrm{t}} \end{bmatrix}。$$

$\boldsymbol{\Psi}_{\mathrm{repair}}$ 必须为 $d \times d$ 可逆矩阵，以保证新节点能够通过左乘 $\boldsymbol{\Psi}_{\mathrm{repair}}^{-1}$ 获得 $\boldsymbol{M}\boldsymbol{\psi}_f$。又因为 $\boldsymbol{M}$ 为对称矩阵，所以 $(\boldsymbol{M}\boldsymbol{\psi}_f)^{\mathrm{t}} = \boldsymbol{\psi}_f{}^{\mathrm{t}}\boldsymbol{M}$，从而在新节点可以恢复失效节点中所存储的数据。

2.2 双线性配对假设

本文所提方案的安全性是建立在判定性 BDHE 假设上的。将在群 G 上的判定性 BDHE 假设定义如下。

定义 1 判定性 BDHE 假设[18]。令 G 和 G_T 为 2 个素数 p 阶群，e 为双线性映射：$G \times G \to G_T$，g 为群 G 的生成元。令 β，$\gamma \leftarrow \mathbb{Z}_p$，$b \leftarrow \{0,1\}$。如果

$b=0$，令 $Z=e(g,g)^{\beta^{n+1}\gamma}$；否则令 $Z\leftarrow G_T$。那么该问题的实例由

$$\{g^{\gamma},Z\}\cup\{g^{\beta^i}:i\in[0,n]\cup[n+2,2n]\}$$

组成。该问题的难题是猜测 b。令攻击者为 A，若 A 能够得到 b，则认为攻击成功。将 A 获取的 b 的优势定义为 $\mathrm{AdvBDHE}_{A,n}(\lambda)=\left|Pr[A\ \ wins]-\frac{1}{2}\right|$。判定性 BDHE 假设为对于任意概率多项式时间内的攻击者 A，其优势是可忽略的。

2.3　基于双线性配对的门限理论简述

令 G 为素数 p 的有限循环群，g 为群 G 的生成元。基于双线性配对的门限机制中，分享者拥有秘密 g^x，以及 $g^{a_1},\cdots,g^{a_{t-1}}$，$t$ 个共享者可以利用分享者分发给他们的分享秘密重建秘密 g^x。令随机选取的多项式为

$$F(\alpha)=g^{x+a_1\alpha+\cdots+a_{t-1}\alpha^{t-1}}\bmod p$$

分享者分发给共享者的秘密为

$$F(k)=g^{f(k)}=g^x(g^{a_1})^k\cdots(g^{a_{t-1}})^{k^{t-1}}$$

任意 t 个共享者可以利用式

$$g^x=F(0)=\prod_{k\in A}g^{x_k\lambda_k}=\prod_{k\in A}F(k)^{\prod_{l\in A}^{l\neq k}\frac{l}{l-k}}$$

获取秘密 g^x。

3　部分适应性安全 RC(SRCS)编码

3.1　SRCS 系统模型

图 1 为存储系统模型示意。SRCS 的存储系统主要由 2 组服务器组成，一组为存储服务器，用以存储经过编码的数据片段，称为存储节点，用 H_i $(1\leqslant i\leqslant n)$表示，其中 H_F 代表在存储过程中失效的存储节点，H_R 代表用以恢复 H_F 失效的存储节点；一组为密钥服务器，用以存储和分发算法中所需要的密钥，用 K_i $(1\leqslant i\leqslant m)$表示。

用以实现安全编码的数据所有者密钥为(upk，usk)，其中 $upk=g^{usk}$ 为数据所有者的公钥，usk 为数据所有者的私钥。令 G 为素数 p 阶有限循环群，g 为群 G 的生成元，随机选取多项式 $f(\alpha)$，利用门限思想生成 usk 的密钥片段并分散存储于密钥服务器中，且可以利用大于等于 t 个密钥片段获取 usk。同时定义 G_T 也为 p 阶有限循环群，$e:G\times G\rightarrow G_T$ 为高效不退化双线性配对。

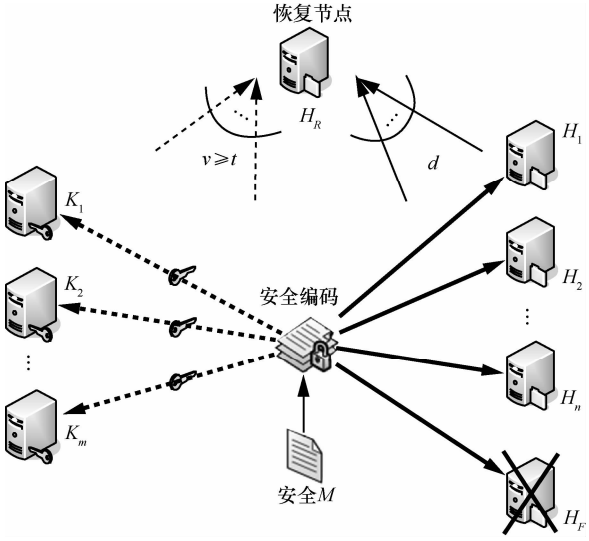


图 1　存储系统模型

在对数据进行编码存储时，存储系统将数据构造为消息矩阵 M，并利用 upk 和编码矩阵 $\boldsymbol{\Psi}$ 对 M 进行编码，并获取 n 个数据片段，并分别存储于 n 个存储节点。当存储节点 H_F 失效时，新的存储节点 H_R 需要先从大于等于 t 个密钥服务器中获取密钥片段，并得到 usk 的相关信息。最后存储系统利用 usk 和恢复矩阵 $\boldsymbol{\Psi}_{\mathrm{repair}}$ 以及消息矩阵获取 H_F 中存储的数据片段。图 1 中粗实线为数据块分散存储过程，粗虚线为密钥共享过程，细实线为恢复节点从 d 个存储服务器获取数据的过程，细虚线为恢复节点从 $v\geqslant t$ 个密钥服务器获取共享秘密的过程。

3.2　SRCS 流程

SRCS 的安全性是建立在确定性 BDHE 假设的基础上的，下面为具体步骤。

Step1　系统初始化。

1) 密钥系统安全参数生成

令 $PairGen$ 为参数生成算法，其输入为安全参数 1^{λ}，输出为四元组 $\delta=(p,G,G_T,e)$。随机选取 $h_1,\cdots,h_m\in G$，令系统安全参数为 $\pi=(\delta,g,h_1,\cdots,h_m,t,m)$。

2) 密钥系统公私钥生成

随机选择 $x_1,\cdots,x_m\in Z_p$，和 $t-1$ 阶多项式 $f\in Z_p[\alpha]$，且 $f(0)=x$，计算

$$X=e(g,g)^x$$

令系统公钥为

$$MPK = X$$

系统私钥为

$$MSK = \langle x, f \rangle$$

3) 密钥服务器系统构建。系统为每个密钥服务器生成一对公私钥 (kpk_i, kdk_i)，假设多项式 f 为

$$f(\alpha) = x + a_1\alpha + \cdots + a_{t-1}\alpha^{t-1} \bmod p$$

计算 $S_i = g^{f(i)}$，随机选取 $r_i \in Z_p$，令第 i 个密钥服务器的公钥 kpk_i 为 i，私钥 kdk_i 为

$$(g^{-r_i}, h_1^{r_i}, \cdots, h_{i-1}^{r_i}, g^{f(i)}h_i^{r_i}, h_{i+1}^{r_i}, \cdots, h_m^{r_i})$$

Step2 数据编码存储过程。

编码存储过程分为 2 个部分，第一部分为存储服务器存储编码数据的过程，第二部分为密钥服务器分散存储编码密钥的过程。

1) 存储服务器存储过程

令 $H: \{0,1\}^* \to G$ 为 Hash 算法，$h_{ID} = H(M)$，将编码后的数据表示为 C，存储在第 i 个存储服务器中的数据表示为 C_i，则有

$$\begin{aligned} C &= (A, \beta, B) \\ &= (g^r, h_{\mathrm{ID}}, (\boldsymbol{\Psi} \times \boldsymbol{M}) \cdot e(upk, h_{\mathrm{ID}}{}^r)) \\ C_i &= (A_i, \beta, B_i, \boldsymbol{\psi}_i^{\mathrm{t}}) = (A, \beta, B_i, \boldsymbol{\psi}_i^{\mathrm{t}}) \\ &= (g^r, h_{\mathrm{ID}}, (\boldsymbol{\psi}_i^{\mathrm{t}} \times \boldsymbol{M}) \cdot e(upk, h_{\mathrm{ID}}{}^r), \boldsymbol{\psi}_i^{\mathrm{t}}) \end{aligned}$$

其中，$r \in Z_p$，令 $\boldsymbol{M}_i = \boldsymbol{\psi}_i^{\mathrm{t}} \times \boldsymbol{M}$。

2) *usk* 共享过程

令 R 代表共享 *usk* 的密钥服务器集合。随机选取 $\gamma \in Z_p$，系统计算

$$Hdr = (c_1, c_2) : c_1 = g^{\gamma}, c_2 = (\prod_{j \in R} h_k)^{\gamma}$$

令 $usk = e(g,g)^{x\gamma}$，系统生成 $\langle Hdr, usk \rangle$，并将 $\langle R, Hdr \rangle$ 发送给共享密钥服务器。

对于密钥服务器 $i \in R$，其可以使用所得到的 Hdr 和 kdk_i 来计算 σ_i，为

$$\begin{aligned} & e\left(g^{f(i)}h_i^{r_i}\prod_{j \in R\{i\}} h_j^{r_i}, c_1\right) e(g^{-r_i}, c_2) \\ &= e\left(g^{f(i)}\left(\prod_{j \in R} h_j\right)^{r_i}, g^{\gamma}\right) e\left(g^{-r_i}, \left(\prod_{j \in R} h_j\right)^{\gamma}\right) \\ &= e(g,g)^{f(i)\gamma} = \sigma_i \end{aligned}$$

其所存储的共享秘密为 $\xi_i = (h_{\mathrm{ID}}, h_{\mathrm{ID}}{}^{\sigma_i})$。

Step3 失效节点恢复过程。

令失效节点中存储的内容为 $C_f = (g^r, h_{\mathrm{ID}}, (\boldsymbol{\psi}_f{}^{\mathrm{t}} \times \boldsymbol{M}) \cdot e(upk, h_{\mathrm{ID}}{}^r), \boldsymbol{\psi}_f{}^{\mathrm{t}})$，恢复节点表示为 H_R，H_R 从大于等于 t 个密钥服务器中获取 ξ_{i_j}，其中，$i_1 \neq i_2 \neq \cdots \neq i_t$，并令集合 $S = \{i_1, \cdots, i_t\}$，并从 d 个存储服务器中获取 C_j，其中 $j_1 \neq j_2 \neq \cdots \neq j_d$，从而有

$$\begin{aligned} h_{\mathrm{ID}}{}^{usk} &= h_{\mathrm{ID}}{}^{\prod_{i \in S} \sigma_i^{\lambda_i}} = h_{\mathrm{ID}}{}^{\prod_{i \in S} (e(g,g)^{f(i)\gamma})^{\lambda_i}} \\ &= h_{\mathrm{ID}}{}^{\prod_{i \in S} e(g,g)^{f(i)\lambda_i\gamma}} = h_{\mathrm{ID}}{}^{e(g,g)^{f(0)\gamma}} \\ &= h_{\mathrm{ID}}{}^{e(g,g)^{x\gamma}} \end{aligned}$$

其中，$\lambda_i = \prod_{j \in S}^{j \neq i} j \big/ j - i$ 为拉格朗日系数。

$$M_f = \frac{B_f}{e(A, h_{\mathrm{ID}}{}^{usk})} = \frac{M_f e(upk, h^r)}{e(g^r, h_{\mathrm{ID}}{}^{usk})}$$

$$\boldsymbol{\Psi}_{\mathrm{repair}} = [\boldsymbol{\psi}_{j_1}^{\mathrm{t}}, \cdots, \boldsymbol{\psi}_{j_d}^{\mathrm{t}}]^{\mathrm{t}}$$

$$M_R = (\boldsymbol{\Psi}_{\mathrm{repair}}{}^{-1} M_f)^{\mathrm{t}}$$

H_R 中存储的数据为 $C_R = (g^r, h_{\mathrm{ID}}, (\boldsymbol{\psi}_f{}^{\mathrm{t}} \times \boldsymbol{M}_R) \cdot e(upk, h_{\mathrm{ID}}{}^r), \boldsymbol{\psi}_f{}^{\mathrm{t}})$。

4 分析

4.1 安全性分析

在 SRCS 中，用户的数据实际上被封装了 2 次，第一次是利用编码的方法将数据编码为一个 $n \times d$ 矩阵，第二次是利用用户的私钥将矩阵加密，并将用户私钥共享给密钥服务器用以容错修复。系统若需要对某一节点进行恢复，新节点必须能够从共享密钥服务器上获取足够多的信息才能够获取用户私钥，从而获取消息恢复节点。这就极大降低了攻击者利用新节点并串谋部分密钥服务器获取存储信息的可能。由此可见，SRCS 的安全性是建立在基于门限的密码体制安全性的基础上，为了证明 SRCS 的安全性，将针对系统中的门限体制构建如下攻击游戏。

定义 2 部分适应性攻击。假设存在使用 SRCS 的存储系统 T，其密钥服务器集合为 K。攻击者在系统构造之前已经控制了一个密钥服务器集合 $\zeta \subset K$。攻击过程中，攻击者可以询问任意服务器 $k \in K$ 且 $k \notin \zeta$ 的解密密钥，并可以询问 $t-1$ 次服务器 $k \in \zeta$ 的解密密钥。攻击者任意选取服务器集合 $\theta \subset \zeta$ 中的密文发起挑战。

由定义 1 可以看出，部分适应性攻击的攻击强

度低于适应性攻击，却要高于静态攻击，因为攻击者可以适应性地选择集合 ζ 。

部分适应性攻击游戏：假定算法 B 是攻击者 A 用来打破判定性 BDHE 假设的实例，进而攻击 SRCS 的算法。B 将被提供一个判定性 BDHE 假设的挑战。利用该挑战，B 可以模拟系统参数、系统公钥、A 控制的密钥服务器的解密密钥以及 A 询问获得的密文。A 无法区分出模拟数据与真实数据的区别，也无法获知其正与模拟器进行交互。B 使用 A 的猜测来解决判定性 BDHE 假设。因为判定性 BDHE 假设是正确的，所以不存在算法 B，而 A 也就不可能实现对 SRCS 中加密体系的攻击，SRCS 是安全的。

初始化：B 获取包含有 g^{γ}、Z 以及集合 $\{g^{\beta^i}: i\in[0,n]\cup[n+2,2n]\}$ 的 BDHE 挑战实例。攻击者 A 可以控制一个密钥服务器集合 R'。A 可以得到系统参数、系统公钥、集合 R' 以外的被攻击密钥服务器的解密密钥以及最多 $t-1$ 个集合 R' 内密钥服务器的解密密钥。A 可以随时发起对上述内容的询问，B 按照统一的方式回应询问。B 输出系统参数 $\pi=(\delta,g,h_1,\cdots,h_n,t,n)$。

公钥模拟： B 可以通过计算获取系统公钥 X，密钥服务器的公钥为其序号 i。

被攻击服务器解密密钥模拟：B 构建多项式 $f(\alpha)$，并随机选取一个拥有 $t-1$ 个密钥服务器子集 $A^*\subseteq R'$。B 计算密钥服务器 $i\in A^*$ 的解密密钥。如果攻击者询问集合 R'/A^* 中服务器的解密密钥，B 将会返回失败消息表示其无法获知。同时 B 可利用 BDHE 挑战实例计算获得密钥服务器 $i\notin R'$ 的解密密钥。

挑战：A 选择攻击服务器集合 $R^*\subset R'$。B 计算 $\langle Hdr,sk\rangle$，并将 $\langle Hdr,sk\rangle$ 发送给 A。

猜测：A 输出 b'，B 接收 b' 并将其发送给 BDHE 挑战者。

根据上述游戏可以得到如下定理。

定理 1　如果部分适应性攻击者 A 在 τ 时间内拥有优势 ε 来打破判定性 BDHE 假设，那么一定存在算法 B 能够在 τ' 时间内拥有优势 ε' 来打破判定性 BDHE 假设。其中，$\varepsilon'\geqslant\dfrac{1}{C_n^{t-1}}\varepsilon$，$\tau'\leqslant\tau+O(1)\tau_{\text{pair}}+O(n^2)\tau_{\text{Exp}}$，$\tau_{\text{pair}}$ 代表配对运算的消耗，τ_{Exp} 代表求幂运算的时间复杂度。

证明　第 1 阶段：初始化。B 收到包含有 g^{γ}、Z 以及集合 $\{g^{\beta^i}: i\in[0,n]\cup[n+2,2n]\}$ 的 BDHE 挑战实例。攻击者 A 可以控制一个密钥服务器集合 $R'\subseteq[1,n]$。A 可以得到系统参数、系统公钥、集合 R' 以外的被攻击密钥服务器的解密密钥以及最多 $t-1$ 个集合 R' 内密钥服务器的解密密钥。A 可以随时发起对上述内容的询问，B 按照统一的方式回应询问。B 生成 $y_0,\cdots,y_n\leftarrow Z_p$，令 $h_i=g^{y_i}\ (i\in R')$，$h_i=g^{y_i+\beta^i}\ (i\in[1,n]/R')$。$B$ 输出系统参数 $\pi=(\delta,g,h_1,\cdots,h_n,t,n)$。

第 2 阶段：公钥模拟。定义 $x=y_0\beta^{n+1}$，B 无法获得 x。B 可以通过计算 $X=e(g,g)^x=e(g^{\beta},g^{\beta^n})^{y_0}$ 获取系统公钥 X，密钥服务器的公钥为其序号 i。

第 3 阶段：被攻击服务器解密密钥模拟。B 构建多项式

$$f(\alpha)=x+a_1\alpha+\cdots+a_{t-1}\alpha^{t-1}\bmod p \tag{1}$$

其中，$x,a_1,\cdots,a_{t-1}\in Z_p$。$B$ 随机选取一个拥有 $t-1$ 密钥服务器子集 $A^*\subseteq R'$。对任意 $k\in A^*$，B 随机选取 $S_k\in Z_p$，并令 $f(k)=S_k$，$f(0)=x$。依据拉格朗日插值公式，$f(\alpha)$ 可表示为

$$\begin{aligned}f(\alpha)&=\sum_{k\in A^*\cup\{0\}}f(k)\left(\prod_{l\in A^*\cup\{0\}}^{l\neq k}\frac{l-\alpha}{l-k}\right)\\&=\prod_{l\in A^*}\frac{l-\alpha}{l}x+\sum_{k\in A^*}f(k)\left(\frac{\alpha}{k}\prod_{l\in A^*}^{l\neq k}\frac{l-\alpha}{l-k}\right)\end{aligned} \tag{2}$$

再令

$$f(\alpha)=\prod_{l\in A^*}\frac{l-\alpha}{l}x+f'(\alpha) \tag{3}$$

其中，

$$f'(\alpha)=\sum_{k\in A^*}f(k)\left(\frac{\alpha}{k}\prod_{l\in A^*}^{l\neq k}\frac{l-\alpha}{l-k}\right)$$

B 随机选取 $r_i\in Z_p$，可计算密钥服务器 $i\in A^*$ 的解密密钥 kdk_i 为

$$(g^{-r_i},h_1^{r_i},\cdots,h_{i-1}^{r_i},g^{f(i)}h_i^{r_i},h_{i+1}^{r_i},\cdots,h_n^{r_i}) \tag{4}$$

如果攻击者询问集合 R'/A^* 中服务器的解密密钥，B 将会返回失败消息表示其无法获知，该事件发生的概率为 $\dfrac{1}{C_{|R'|}^{t-1}}\geqslant\dfrac{1}{C_n^{t-1}}$。

为了计算密钥服务器 $i\notin R'$ 的解密密钥 kdk_i，B

随机选取 $z_i \leftarrow Z_p$，令 $r_i = \prod_{l \in A^*} \frac{l-i}{l}\left(z_i - y_0 \beta^{n+1-i}\right)$。那么

$$kdk_i = (d_{i,0}, \cdots, d_{i,n}):$$
$$d_{i,0} = g^{-r_i}, d_{i,i} = g^{f(i)} h_i^{r_i}, d_{i,j} = h_j^{r_i} (\forall j \neq i)$$

可以看出，上述元素都可以利用 BDHE 挑战实例和式(3)计算获得。其中，

$$\begin{aligned} d_{i,i} &= g^{f(i)} h_i^{r_i} \\ &= g^{\prod_{l \in A^*} \frac{l-i}{l}\left(y_0 \beta^{n+1}\right) + f'(i)} \prod_{l \in A^*} h_i^{\frac{l-i}{l}\left(z_i - y_0 \beta^{n+1-i}\right)} \\ &= g^{\prod_{l \in A^*} \frac{l-i}{l}\left(y_0 \beta^{n+1}\right) + f'(i)} \left(g^{y_i + \beta^i}\right)^{\prod_{l \in A^*} \frac{l-i}{l}\left(z_i - y_0 \beta^{n+1-i}\right)} \\ &= g^{\prod_{l \in A^*} \frac{l-i}{l}\left(y_i (z_i - y_0 \beta^{n+1-i}) \beta^i z^i\right) + f'(i)} \end{aligned}$$

因此 B 可以回应所有 A 针对 $i \notin R'$ 的服务器解密密钥的询问。

第 4 阶段：挑战。A 选择攻击服务器集合 $R^* \subset R'$。B 计算

$$Hdr = (c_1, c_2): c_1 = g^{\gamma}, c_2 = \left(\prod_{j \in R^*} h_j\right)^{\gamma}, \quad sk \leftarrow Z^{y_0}$$

并将 $\langle Hdr, sk \rangle$ 发送给 A。

B 可以利用给定的 BDHE 实例来计算 Hdr 和 sk。其中 c_1 和 sk 可以直接从实例中获取。对所有 $i \in R^*$，B 可以获取 $\log_g(h_i)$，所以可得

$$c_2 = \left(\prod_{j \in R^*} h_j\right)^{\gamma} = \left(\prod_{j \in R^*} g^{y_j}\right)^{\gamma} = \left(g^{\gamma}\right)^{\sum_{j \in R^*} y_j}$$

第 5 阶段：猜测。A 输出 b'，B 接收 b' 并将其发送给 BDHE 挑战者。

假设 B 在部分适应性攻击游戏中不存在返回失败消息的情况。当 $b=0$ 时，有 $Z = e(g,g)^{\beta^{n+1}\gamma}$，$sk = Z^{y_0} = e(g,g)^{x\gamma}$，$x = y_0 \beta^{n+1}$，所以此时挑战为一个在 γ 下的有效密文，当 $b=1$ 时，攻击者获得 $\langle Hdr, sk \rangle$，对于随机数 γ，Hdr 是有效的，又因为 Z 为 G_T 上的随机数，所以 $sk = Z^{y_0}$ 为 G_T 上的均匀的随机元素。以此可以看出，B 对于判定性 BDHE 假设的优势等同于攻击者 A 攻破 SRCS 中构建的门限密码体制的优势。由此可得，B 的优势为 $\varepsilon' \geqslant \frac{1}{C_n^{t-1}} \varepsilon$。

B 的额外消耗在于需要对 A 的询问进行应答，在第 1 阶段 B 需要在 G 内进行 $n+1$ 次幂运算；在第 2 阶段 B 计算 $e(g^{\beta}, g^{\beta^n})$，需要 1 次幂运算和 1 次配对运算；在第 3 阶段，B 的主要运算消耗最多为 $(t-1)(n+2)+(n-t)(n+t+1)$ $(1 \leqslant t \leqslant n)$ 次幂运算；第 4 阶段 B 需要在 G 内进行 1 次幂运算来计算挑战密文。所以 B 得时间复杂度为 $\tau' \leqslant \tau + O(1)\tau_{\text{pair}} + O(n^2)\tau_{\text{Exp}}$。

4.2 存储开耗分析

为了便于分析，将 G 和 G_T 中元素的比特长度分别定义为 l_1 和 l_2。由于在密钥服务器中仅仅保存共享秘密 $\xi_i = (h_{\text{ID}}, h_{\text{ID}}^{usk_i})$，其中，$h_{\text{ID}} \in G$，所以可得每一个密钥服务器中的存储消耗为 $2l_1$ bit。假设存在 1 bit 消息，其在存储服务器中存储的数据为 $C_i = (A_i, \beta, B_i, \psi_i^{\iota})$，其中 $A_i, \beta \in G$，$B_i \in G_T$，所以 1 bit 消息在每一个存储服务器上的存储消耗为 $2l_1 + d \times \lceil \text{lb}p \rceil \times (l_2 + 1)$ bit。

5 结束语

本文提出了一种安全的 RC 编码——SRCS 编码，它将基于公钥的门限体制与 RC 编码相结合，在保证了原有 RC 编码特性的基础上确保了容错过程中的安全性，解决了目前的 RC 编码中存在的安全问题，通过证明可以看出，SRCS 可以抵抗部分适应性选择密文攻击。在下一步工作中将主要解决如何实现抵抗适应性选择密文攻击的问题，进一步提高编码的安全性。

参考文献：

[1] DEAN J. Experiences with mapreduce, an abstraction for large-scale computation[A]. PACT 2006[C]. Seattle, 2006.16-20.

[2] WEATHERSPOON H, KUBIATOWICZ J D. Erasure coding vs replication: a quantitative comparison[A]. The 1st International Workshop on Peer to Peer Systems (IPTPS)[C]. Cambridge, MA, USA, 2002.1-6.

[3] WU Y N, DIMAKIS A G , RAMCHANDRANY K. Deterministic regenerating codes for distributed storage[A]. Allerton Conference on Control, Computing and Communication, Urbana-Champaign[C]. Allerton House, Illinois, USA, 2007.1-8.

[4] WU Y N. Existence and construction of capacity-achieving network codes for distributed storage[J]. IEEE Journal on Selected Areas in Communications, 2010, 28 (2):277-288.

[5] DIMAKIS A G, ALEXANDROS G. A survey on network codes for distributed storage[J]. Computing Research Repository, 2011, 99 (3): 476-489.

[6] HU Y C, XU Y L, WANG X Z. MCR: a mutual cooperative recovery mechanism in peer-to-peer storage systems[A]. The 2nd IEEE International Conference on Broadband Network & Multimedia Technology[C].

Hefei, China, 2009.681-686.

[7] HU Y C, XU Y L, WANG X Z, *et al*. Cooperative recovery of distributed storage systems from multiple losses with network coding[J]. IEEE Journal on Selected Areas in Communications, 2010, 28(2): 268-276.

[8] SHUM K W, HU Y C. Exact minimum-repair-bandwidth cooperative regenerating codes for distributed storage systems[A]. IEEE International Symposium on Information Theory[C]. Saint-Petersburg, Russia, 2011.1442-1446.

[9] SHUM K W. Cooperative regenerating codes for distributed storage systems[A]. IEEE International Conference on Communications (ICC)[C]. Hong Kong, China, 2011.1-5.

[10] KERMARREC A M, SCOUARNECY N L, STRAUBY G. Repairing multiple failures with coordinated and adaptive regenerating codes[A]. In International Symposium on Network Coding (NetCod)[C]. Rennes, France, 2011.1-6.

[11] LI J, WANG X, LI B C. Pipelined regeneration with regenerating codes for distributed storage systems[A]. International Symposium on Network Coding (NetCod)[C]. Rennes, France, 2011.1-6.

[12] GKANTSIDIS C, RODRIGUEZ P R. Cooperative security for network coding file distribution[A]. The 25th IEEE International Conference on Computer Communications[C]. Barcelona, Spain, 2006.1-13.

[13] LI Q M, LUI J C S. On the security and efficiency of content distribution via network coding[J]. IEEE Transactions on Dependable and Secure Computing, 2012, 9(2): 211-221.

[14] LIN H Y, TZENG W G. A secure decentralized erasure code for distributed networked storage[J]. IEEE Transactions on Parallel and Distributed Systems, 2010, 21 (11): 1586-1594.

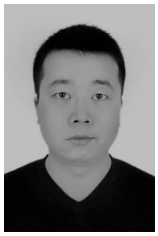
[15] DIMAKIS A G , PRABHAKARAN V, RAMCHANDRAN K. Decentralized erasure codes for distributed networked storage[J]. IEEE Trans Information Theory, 2006, 52 (6): 2809-2816.

[16] QIN B, WU Q H. Provably secure threshold public-key encryption with adaptive security and short ciphertexts[J]. Information Sciences, 2012, 210:67-80.

[17] RASHMI K V, SHAH N B. Optimal exact-regenerating codes for distributed storage at the MpSR and MBR points via a product-matrix construction[J]. IEEE Transactions on Information Theory, 57(8): 5227-5239.

[18] BONEH D. Hierarchical identity based encryption with constant size ciphertext[A]. EUROCRYPT[C]. Heidelberg, 2005.440-456.

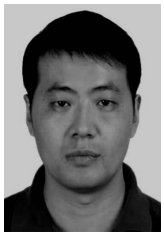
作者简介：



谭鹏许（1984-），男，河南许昌人，解放军信息工程大学博士生，主要研究方向为网络安全、云存储。

陈越（1965-），男，河南开封人，博士，解放军信息工程大学教授、博士生导师，主要研究方向为网络与信息安全。

兰巨龙（1962-），男，博士，国家数字交换系统工程技术研究中心教授、博士生导师，主要研究方向为网络路由理论与技术、并行交换结构和 IPv6 技术。



贾洪勇（1975-），男，河南西平人，博士，解放军信息工程大学讲师，主要研究方向为应用密码学、网络安全。