

1. 在腾讯会议系统中，对称密码技术和公钥密码技术适合应用在哪几个阶段？说明理由。
2. 假设计算能力遵循摩尔定律，分析三重DES目前在计算上是否安全。
3. 简述散列函数 MD5 的碰撞问题。既然 MD5 存在碰撞问题，为何 <http://mirrors.ustc.edu.cn/ubuntu-releases/16.04/> 仍然给出 MD5 值作为完整性验证的依据。
4. 简述用 RSA 公钥算法实现数字签名的过程。

T1

对称密码技术：用于会议过程中音视频数据传输阶段的加密，因为对称加密速度快，适合对大量实时数据进行加密

公钥密码技术：用于会议开始时的密钥交换与身份认证阶段，因为公钥算法安全性高但速度慢，适合加密会话密钥或验证身份

T2

按摩尔定律增长，目前 3DES 在计算上仍安全

分析：

1999年，DES的密钥长度仅为56比特，破解密文需要 2^{56} 次穷举搜索

现在2025年，过去26年，根据摩尔定律每18个月计算能力提升一倍，大约17.33个周期，提升 $2^{17.33}$ ，大约160000倍

而3DES破解密文需要 2^{112} 次穷举搜索，对比DES提升 2^{56} 倍，显著高于计算能力的提升

T3

用于消息鉴别的散列函数H必须满足下面所有条件才不会存在碰撞问题：

- 对于任意给定的值h，要找到一个x满足 $H(x)=h$ ，在计算上是不可能的
- 对于任意给定的数据块x，要找到一个y*x并满足 $H(y)=H(x)$ ，在计算上是不可能的
- 要找到一对(x, y)满足 $H(y)=H(x)$ ，在计算上是不可能的

而 MD5的散列码长度为128比特，已经被证明存在碰撞问题，并被找出实例

Sequence #1															
d1	31	dd	02	c5	e6	ee	c4	69	3d	9a	06	98	af	f9	5c
2f	ca	b5	87	12	46	7e	ab	40	04	58	3e	b8	fb	7f	89
55	ad	34	06	09	f4	b3	02	83	e4	88	83	25	71	41	5a
08	51	25	e8	f7	cd	c9	9f	d9	1d	bd	f2	80	37	3c	5b
d8	82	3e	31	56	34	8f	5b	ae	6d	ac	d4	36	c9	19	c6
dd	53	e2	b4	87	da	03	fd	02	39	63	06	d2	48	cd	a0
e9	9f	33	42	0f	57	7e	e8	ce	54	b6	70	80	a8	0d	1e
c6	98	21	bc	b6	a8	83	93	96	f9	65	2b	6f	f7	2a	70
Sequence #2															
d1	31	dd	02	c5	e6	ee	c4	69	3d	9a	06	98	af	f9	5c
2f	ca	b5	07	12	46	7e	ab	40	04	58	3e	b8	fb	7f	89
55	ad	34	06	09	f4	b3	02	83	e4	88	83	25	f1	41	5a
08	51	25	e8	f7	cd	c9	9f	d9	1d	bd	72	80	37	3c	5b
d8	82	3e	31	56	34	8f	5b	ae	6d	ac	d4	36	c9	19	c6
dd	53	e2	34	87	da	03	fd	02	39	63	06	d2	48	cd	a0
e9	9f	33	42	0f	57	7e	e8	ce	54	b6	70	80	28	0d	1e
c6	98	21	bc	b6	a8	83	93	96	f9	65	ab	6f	f7	2a	70
Both produce MD5 digest 79054025255fb1a26e4bc422aef54eb4															

但对于一般文件下载、软件镜像校验等场景，其安全性要求并不是极高，因此仍可用于验证消息完整性

T4

签名过程：

1. 发送方 A 对原始消息 M 计算摘要 $H(M)$
2. A 用自己的私钥对摘要加密，生成数字签名
3. 将消息和签名一起发送给接收方 B

验证过程：

1. 接收方 B 用 A 的 公钥对签名解密得到摘要 $H_1(M)$
2. B 再计算接收到消息的摘要 $H_2(M)$
3. 若 $H_1(M)=H_2(M)$ ，则证明消息未被篡改且确系A发送