论公钥协议的安全性

DANNY DOLEV 与 ANDREW C. YAO，IEEE 会员

摘要——近年来，使用公钥加密来提供安全的网络通信受到了广泛关注。这类公钥系统通常能有效抵御"被动"窃听者，即仅窃听线路并试图解密消息的人。然而，正如 Needham 和 Schroeder 所指出的，设计不当的协议可能容易受到"主动"破坏者的攻击，这种人可能冒充其他用户或篡改传输中的消息。本文提出了几种模型，能够精确地讨论协议的安全性。并给出了在这些模型中用于确定协议安全性的算法和特征描述。

一、引言

利用公钥加密 [1], [11] 来提供安全的网络通信已经受到了广泛关注 [2], [7], [8], [10]。这类公钥系统通常对"被动"窃听者非常有效，即那些仅窃听通信线路并试图解密截获消息的人。然而，正如 Needham 和 Schroeder [8] 所指出的，设计不当的协议可能容易受到"主动"破坏者的攻击，这种人可能冒充其他用户，并可能修改或重放消息。由于一个协议可能以复杂的方式被攻破，那些断言协议安全性的非形式化论证容易出错。因此，需要一个形式化模型来精确地讨论安全性

稿件收到日期：1981年7月15日；修订日期：1982年8月8日。本工作部分由 ARPA 资助（批准号 MDA-903-80-C-102）和 NSF 资助（批准号 MCS-77-05313-A01）。本文部分内容曾于1981年10月28–30日在田纳西州纳什维尔举行的第22届 IEEE 计算机科学基础年会上发表。

D. Dolev 曾就职于斯坦福大学计算机科学系，斯坦福，CA。现就职于希伯来大学数学与计算机科学研究所，耶路撒冷，以色列。

A. C. Yao 就职于斯坦福大学计算机科学系，斯坦福，CA 94305。

问题。我们引入的模型将使我们能够以极少关于破坏者行为的假设，来研究协议族的安全性问题。

我们简要回顾一下公钥加密的要点（更多信息见 [1], [11]）。在公钥系统中，每个用户 X 有一个加密函数 E_{x} 和一个解密函数 D_{x}，两者都是从 $\{0, 1\}^{*}$（所有有限二进制序列的集合）到 $\{0, 1\}^{*}$ 的映射。一个安全的公共目录包含所有 (X, E_{x}) 对，而解密函数 D_{x} 仅用户 X 知道。对 E_{x}, D_{x} 的主要要求是：

1) $E_{x}D_{x} = D_{x}E_{x} = 1$，并且
2) 知道 $E_{x}(M)$ 和公共目录不会泄露关于 M 值的任何信息。

因此，任何人都可以发送 X 一条消息 $E_{x}(M)$，X 将能够通过计算 $D_{x}(E_{x}(M)) = M$ 来解码它，但即使 $E_{x}(M)$ 对他们可用，除 X 以外的任何人都无法找到 M。

我们主要关注在两个用户之间传输秘密明文 M 的协议。为了说明破坏者可能攻破系统的方式，我们考虑几个例子。网络中各方之间发送的消息包含三个字段：发送者姓名、接收者姓名和正文。正文是消息的加密部分。我们将按以下格式书写消息：发送者姓名，正文，接收者姓名。

示例 1：考虑以下在 A 和 B 之间发送明文 M 的协议：
a) A 发送 B 消息 $(A, E_B(M), B)$，
b) B 用消息 $(B, E_A(M), A)$ 回复 A。

这个协议很容易被破坏者 Z 以下列方式攻破：
1) Z 在步骤 a) 中拦截从 A 发送到 B 的消息。
2) Z 向 B 发送消息 $(Z, E_B(M), B)$。
3) B 根据协议（步骤 b)) 用 $(B, E_Z(M), Z)$ 回复 Z。
4) Z 解码 $E_Z(M)$ 以找到明文 M。

克服上述协议弱点的一种方法是在加密文本中同时对发送者姓名和明文进行编码。考虑 Needham 和 Schroeder [8] 中建议的协议的以下变体。

示例 2：考虑以下协议（MA 表示 M 和 A 的连接）：
a) A 发送 B 消息 $(A,E_B(MA),B)$
b) B 通过发送 $(B, E_A(MB), A)$ 回复 A。

我们将在本文后面证明该协议对于破坏者的任意行为都是安全的。如果试图通过添加另一层加密来改进上述协议，会发生什么？

示例 3：考虑以下协议：
a) A 发送 B 消息 $(A,E_B(E_B(M)A),B)$
b) B 通过发送 $(B, E_A(E_A(M)B), A)$ 回复。

令人惊讶的是，这个协议可以通过以下方式被攻破：
1) Z 获取步骤 b) 中从 B 发送回 A 的消息，即 $(B, E_A(E_A(M)B), A)$。
记 $E_A(M)B$ 为 $\tilde{M}$，则 Z 可以从上述消息中提取出 $E_A(\tilde{M})$。
2) Z 发起与 A 的对话，根据协议（步骤 a)) 发送

3) A 作为接收者，用以下消息回复 Z：

4) Z 从步骤 3) 收到的消息中解码出 $\tilde{M}$。由于 $\tilde{M} = E_A(M)B$，Z 现在拥有 $E_A(M)$。
5) Z 建立一个新的连接并向 A 发送消息

6) 现在 A 应该回复 $(A, E_z(E_Z(M)A), Z)$。
7) 在此步骤中，Z 能够找到明文 M。

精确的数学模型将在后续章节中定义。下面我们列出了我们希望建模的系统的基本假设。

1) 在一个完美的公钥系统中，
  a) 使用的单向函数是不可破解的；
  b) 公共目录是安全的，不能被篡改；
  c) 每个人都有权访问所有 E_{x}；
  d) 只有 X 知道 D_x。
2) 在一个两方协议中，只有希望通信的两个用户参与传输过程；不需要第三方在解密或加密方面提供协助。
3) 在一个统一协议中，每个希望通信的用户对使用相同的格式。在前面给出的三个示例中，用户名 A, B 是符号参数，可以是任何两个名字。
4) 关于破坏者的行为，我们将重点关注那些"主动"窃听者。即首先窃听通信线路以获取消息，然后尽其所能试图发现明文的人。更准确地说，我们假设破坏者具有以下能力：
  a) 他可以获取通过网络传递的任何消息。
  b) 他是网络的合法用户，因此特别是可以发起与任何其他用户的对话。

c) 他有机会成为任何用户 A 的接收者。（更一般地，我们允许任何用户 B 可能成为任何其他用户 A 的接收者的可能性。）

我们对本文获得的结果进行总结。将建立两种模型。

1) **级联协议**：这些是用户可以使用公钥加密-解密操作来形成消息的协议；然而，可能会应用多层这样的操作符。示例 1 给出了级联协议的一个简单例子。

2) **名字戳协议**：这些是允许用户附加、删除和检查与明文一起加密的名字的协议。名字戳协议也可以包含多层加密（如示例 2 和 3 所示）。

在第 II 节中，我们证明一个级联协议是安全的，当且仅当以下两个条件都满足：
1) 在 X 和 Y 之间传输的消息总是包含一些 E_{x} 或 E_{Y} 的加密函数层；
2) 在生成回复消息时，每个参与者 A（$A = X, Y$）从不单独应用 D_A 而不应用 E_A。

这给出了安全性的简单特征描述，以及一个用于判定给定级联协议是否安全的高效算法。

在第 III 节中，我们给出一个多项式时间算法来判断给定的名字戳协议是否安全。在第 IV 节中，我们考虑破坏者是否可以在不等待其他人发起对话的情况下攻破协议。这对应于之前关于破坏者行为的讨论中仅使用项目 a) 和 b) 的情况。我们给出了第 II 和 III 节中结果对此情况的扩展。

在结束引言时，我们注意到存在其他类型的破坏活动可能挫败公钥协议（或任何协议）的目的。我们建议读者参考 Needham 和 Schroeder [8] 以进行进一步讨论。网络通信中的破坏问题也出现在其他背景下（见 Dolev [3], Pease 等人 [9]）。

二、级联协议

在本节中，我们考虑一类简单的协议，其中用户用来生成消息的唯一操作是加密-解密操作符。我们的目标是分析此类协议对破坏者的安全性。为此，我们必须建立一个形式化模型。我们必须指定 1) 协议的语法，即用户在每个步骤应用什么操作来生成消息，以及 2) 破坏者可用于发现明文的推理规则。

A. 符号

令 Σ 为一个由不同符号组成的有限集。我们用 Σ^{*} 表示由 Σ 中符号组成的所有有限序列的集合；集合 Σ^{*} 也包含空字符串 λ。我们定义 $\Sigma^{+} = \Sigma^{*} - \langle \lambda \rangle$，即 Σ 上所有非空字的集合。字 α 和 β 的连接记为 $\alpha \beta$。设 $\gamma = \alpha \beta$ 是一个字，则 α 称为 γ 的前缀，β 是 γ 的后缀。

公钥操作符的基本属性是 $E_{x}D_{x} = D_{x}E_{x} = 1$，即恒等函数。因此，任何形式为 $\sigma E_{x}D_{x}\sigma^{\prime}$ 的操作符字符串将等价于 $\sigma \sigma^{\prime}$，即对于所有 $P\in \langle 0,1\rangle^{*}$，$(\sigma E_{x}D_{x}\sigma^{\prime})P = (\sigma \sigma^{\prime})P$。我们将说 $\sigma E_{x}D_{x}\sigma^{\prime}$（或 $\sigma D_xE_x\sigma '$）可以约简为 $\sigma \sigma^{\prime}$。对于任何操作符字符串 γ，令 $\gamma |_x$ 表示通过迭代删除所有 $E_{x}D_{x}$ 和 $D_{x}E_{x}$ 对直到无法进一步约简而得到的完全约简字符串。用 $\bar{\gamma}$ 表示通过对系统中所有用户 X 进行完全约简从 γ 得到的字符

串，$\bar{\gamma}$ 是 γ 的约简形式。注意 $\gamma |_x$ 和 $\bar{\gamma}$ 都是唯一的。

为方便起见，我们有时将 D_x 写为 E_x^c，即 E_x 的补。类似地，E_x 也写为 D_x^c。令 $\gamma = a_1 \cdots a_n$ 是一个由 n 个符号组成的字，每个符号是 E 或 D。定义

字 γ^c 是 γ 的补，当 γ 和 γ^c 被视为操作符时，它满足 $\gamma \gamma^c = \gamma^c \gamma = 1$。对于任何字符串 γ，令 $lt(\gamma)$ 为 γ 中的符号集合。

B. 模型

定义 1：一个两方级联协议 T 由一系列有限字符串指定

其中 $t' = t$ 或 $t - 1$。对于每对不同的用户 X 和 Y，令 $\alpha_{i}(X,Y)$、$\beta_{i}(X,Y)$ 表示字符串 $\tilde{\alpha}_{i}$、$\tilde{\beta}_{i}$ 分别将符号 z_{1},z_{2},z_{3},z_{4} 替换为 E_{X},E_{Y},D_{X},D_{Y}。

显然，$\alpha_{i}(X,Y)\in \{E_{X},E_{Y},D_{X}\}^{*}$ 且 $\beta_{i}(X,Y)\in$ $\{E_X,E_Y,D_Y\}^*$。当用户 X 想要向用户 Y 传输一个秘密明文 M 时，他们按照 T 交换消息如下：

X 发送 Y 消息 $\alpha_{1}(X,Y)M$；
Y 对收到的消息应用 $\beta_{1}(X,Y)$ 并将其发送给 X；
X 对收到的消息应用 $\alpha_{2}(X,Y)$ 并将其发送给 Y；
Y 对收到的消息应用 $\beta_{2}(X,Y)$ 并将其发送给 X；
：

注意，协议是统一的，因为对于任何用户 A,B，$\alpha_{i}(A,B)$ 和 $\beta_{i}(A,B)$ 可以通过将 X 替换为 A 和 Y 替换为 B 从 $\alpha_{i}(X,Y),\beta_{i}(X,Y)$ 得到。为方便起见，我们假设 $\tilde{\alpha}_{i}$ 和 $\tilde{\beta}_i$ 使得 $\alpha_{i}(X,Y),\beta_{i}(X,Y)$ 是约简形式。

定义 2：令 T 为由 $\{\tilde{\alpha}_i, \tilde{\beta}_j | 1 \leqslant i \leqslant t, 1 \leqslant j \leqslant t'\}$ 指定的两方级联协议，并令 X, Y 为两个不同的用户。定义

当 X 希望向 Y 发送明文 M 时，交换的消息是 $N_{i}(X,Y)M$，其中 $i = 1,2,\dots ,t + t^{\prime}$。

示例：考虑由 $(\tilde{\alpha}_1 = z_2z_3, \tilde{\beta}_1 = z_1z_4z_1z_1z_4)$ 给出的协议 T。有 $\alpha_{1}(X,Y) = E_{Y}D_{X}$ 和 $\beta_{1}(X,Y) = E_{X}D_{Y}E_{X}E_{X}D_{Y}$。对于明文 M，传输的消息是 $N_{1}(X,Y)M = E_{Y}D_{X}M$ 和 $N_{2}(X,Y)M = E_{X}D_{Y}E_{X}M$。

到目前为止，我们已经讨论了级联协议的语法。我们现在将定义级联协议安全性的概念，即破坏者何时能够推断出在两个用户之间传输的明文 M。我们首先给出一个形式化定义。令 E 为所有 E_A 的集合，D 为所有 D_A 的集合。令 X, Y, Z 表示不同的用户名。

定义 3：令 T 为由 $\{\tilde{\alpha}_i, \tilde{\beta}_j\}$ 指定的两方级联协议。定义

如果存在某个 $\gamma \in (\Sigma_1(Z) \cup \Sigma_2 \cup \Sigma_3)^*$ 使得对于某个 $N_{j}(X,Y)$，有

则称 T 是**不安全的**；否则称 T 是**安全的**。

备注：显然，T 的安全性的定义与 X, Y, Z 的选择无关。

我们现在给出该定义背后的动机。假设 X 试图向 Y 发送明文 M（使用协议 T）。那么他们之间实际传输的消息是 $N_{i}(X,Y)M$（$i = 1,2,\dots$）并且可能落入破坏者 Z 的手中。获取任何 $N_{i}(X,Y)M$，破坏者 Z 有机会通过重复应用以下三种类型的操作符中的任何一种来转换它：

a) 任何 $\sigma \in \Sigma_1(Z)$
b) 任何 $\sigma \in \Sigma_3$：Z 可以发起与用户 B 的明文传输，声称自己是 A，并在第 $(2i - 1)$ 条消息中向 B 发送任何字符串 P；然后 Z 得到 $\beta_i(A, B)P$，有效地将操作符 $\beta_i(A, B)$ 应用于任何选定的 P；
c) 任何 $\sigma \in \Sigma_2$：令 $\sigma = \alpha_i(A, B)$；有可能 A 将来某个时间希望向 B 传输明文；Z 可以拦截 B 回复 A 的第 $(i - 1)$ 条消息，阻止其到达 A，并用任何选定的字符串 P 替换它，并从 A 接收到字符串 $\alpha_i(A, B)P$。

因此，Z 有机会获得字符串 $\gamma N_{i}(X,Y)M$ 对于任何 $\gamma \in (\Sigma_{1}(Z)\cup \Sigma_{2}\cup \Sigma_{3})^{*}$。这意味着，如果对于某个 $\gamma \in (\Sigma_1(Z)\cup \Sigma_2\cup \Sigma_3)^*$，有 $\overline{\gamma N_i(X,Y)} = \lambda$，则 Z 可能推断出 M。

我们希望指出，为了从 P 获得 $\alpha_{i}(A,B)P$，Z 必须等待 A 发起与 B 的对话。这可能会发生，也可能不会发生。因此，我们对安全性的定义是保守的，因为我们关注的是最坏情况的可能性。

C. 安全协议的特征描述

定义 4：令 $\pi \in \{E, D\}^*$ 为一个字符串，A 为一个用户名。如果

则称 π 关于 A 具有**平衡性质**。

正如我们将看到的，平衡性质是安全级联协议固有的。

定义 5：令 X, Y 为两个不同的用户名。一个两方级联协议 $T = \{\tilde{\alpha}_i, \tilde{\beta}_j\}$ 是一个**平衡级联协议**，如果

1) 对于每个 $i \geqslant 2$，$\alpha_{i}(X, Y)$ 关于 X 具有平衡性质，并且
2) 对于每个 $i \geqslant 1$，$\beta_i(X, Y)$ 关于 Y 具有平衡性质。

备注：我们强调 $\alpha_{i}(X,Y),\beta_{j}(X,Y)$ 对于 $i,j\geqslant 1$ 是约简形式。

引理 1：令 Z 为一个用户名，T 为一个平衡级联协议。那么对于 $(\Sigma_1(Z) \cup \Sigma_2 \cup \Sigma_3)^*$ 中的每个字符串 η，$\bar{\eta}$ 关于每个 $A \neq Z$ 具有平衡性质。

证明见附录 I。

我们现在准备陈述并证明本节的主要结果。令 X, Y 为两个不同的用户名。

定理 1：一个两方级联协议 $T = \{\tilde{\alpha}_i, \tilde{\beta}_j\}$ 是安全的，当且仅当

1) $lt(\alpha_{1}(X,Y))\cap \{E_{X},E_{Y}\} \neq \emptyset$，并且
2) T 是平衡的。

证明：令 Z 为一个与 X 和 Y 不同的用户名。
A) 必要性：假设性质 1) 或 2) 中至少有一个不成立。我们将证明 T 是不安全的，即存在 $\gamma \in (\Sigma_1(Z) \cup \Sigma_2 \cup \Sigma_3)^*$ 使得对于某个 i，$\overline{\gamma N_i(X,Y)} = \lambda$。

如果 1) 不成立，则 $\overline{\gamma N_1(X,Y)} = \lambda$ 其中 $\gamma = \alpha_{1}^{c}\in \Sigma_{1}(Z)^{*}$，证明完成。因此我们可以假设 2) 为假，即 T 不平衡。根据定义，要么某个 $\beta_{k}(X,Y)$ 包含 D_Y 但不包含 E_{Y}，要么某个 $\alpha_{i}(X,Y)$（$i\geqslant 2$）包含 D_{x} 但不包含 E_{x}。我们首先将自己限制在前一种情况（β_{k} 包含 D_{Y} 但不包含 E_{Y}）；后一种情况稍后处理。在此限制下，我们将建立以下更强结果。对于任何 $\delta \in ((E_X,E_Y)\cup D)$，存在 $\gamma \in (\Sigma_1(Z)\cup \{\beta_k(Z,X),\beta_k(Z,Y)\})^*$ 使得 $\gamma \delta = \lambda$。证明将通过 r 进行归纳，r 是字符串 δ 中 E_{X} 和 E_{Y} 的数量。

如果 $r = 0$，那么 $\gamma = \delta^c\in (\Sigma_1(Z))^*$ 满足要求。现在令 $r > 0$ 并假设结果对所有更小的 r 值成立。令 δ 为一个恰好包含 r 个 E_{X} 和 E_{Y} 的字符串。不失一般性，我们可以假设最左边的 E 是 E_{Y}。写 $\delta = \sigma_{1}E_{Y}\sigma_{2}$，其中 $lt(\sigma_1)\cap \langle E_X,E_Y\rangle = \emptyset$；显然，$\sigma_1^c\in (\Sigma_1(Z))^*$。根据假设，$\beta_{k}(Z,Y)$ 包含 D_{Y} 但不包含 E_{Y}；因此我们可以写 $\beta_{k}(Z,Y) = \tau_{1}D_{Y}\tau_{2}$，其中 $\tau_{1}\in \{E_{Z},D_{Y}\}^{*}$ 且 $\tau_{2}\in \{E_{Z}\}^{*}$。显然，$\tau_i^c\in (\Sigma_1(Z))^*$ 对于 $i = 1,2$。现在 σ_{2} 包含 $r - 1$ 个 E，根据归纳假设，存在 $\underline{\gamma'}\in (\Sigma_1(Z)\cup \{\beta_k(Z,X),\beta_k(Z,Y)\})^*$ 使得 $\underline{\gamma'\sigma_2} = \lambda$。定义 $\gamma = \gamma'\tau_1^c\beta_k(Z,Y)\tau_2^c\sigma_1^c$。那么根据上述讨论，$\gamma \in (\Sigma_1(Z)\cup \{\beta_k(Z,X),\beta_k(Z,Y)\})^*$。此外，

这完成了归纳步骤。

仍需证明当某个 $\alpha_{i}(X,Y)$ $(i\geqslant 2)$ 包含 D_{x} 但不包含 E_{x} 时，T 是不安全的。可以证明以下更强结果：对于任何 $\delta \in (\{E_X,E_Y\} \cup D)^*$，存在 $\gamma \in (\Sigma_1(Z)\cup \{\alpha_i(X,Z),\alpha_i(Y,Z))\})^*$ 满足 $\gamma \delta = \lambda$。证明与前一个证明几乎相同，不再重复。

B) 充分性：假设性质 1) 和 2) 都满足；我们将证明 T 是安全的。

假设相反，存在 $\gamma \in (\Sigma_{1}(Z) \cup \Sigma_{2} \cup \Sigma_{3})^{*}$ 使得对于某个 i，$\gamma N_{i}(X,Y) = \lambda$。我们将导出一个矛盾。写 $\gamma N_{i}(X,Y) = P\alpha_{1}(X,Y)$ 使得 $P \in (\Sigma_{1}(Z) \cup \Sigma_{2} \cup \Sigma_{3})^{*}$。根据 γ 的定义，我们有

根据协议的定义，$lt(\alpha_{1}(X,Y))\subseteq \{E_{x},D_{x},E_{Y}\}$。我们区分两种情况。

情况 **$B1$**，$E_{Y} \in lt(\alpha_{1}(X,Y))$：由于字符串 α_{1} 不包含 D_{Y}，(1) 式成立的可能性是 $\overline{P}$ 包含一些 D_{Y} 但不包含 E_{Y}。这意味着 P 关于 Y 不具有平衡性质。由于 $Y \neq Z$，这与引理 1 矛盾。

情况 B2，$E_{Y} \notin lt(\alpha_{1}(X,Y))$：在这种情况下 $D_{X} \notin lt(\alpha_{1}(X,Y))$，因为 $\alpha_{1}(X,Y)$ 是约简形式且协议满足引理中的性质 1)。这意味着 $\alpha_{1}(X,Y) = E_{X}^{+}$。类似于情况 B1，(1) 式成立的可能性是 $\overline{P}$ 包含 D_{X} 但不包含 E_{X}，这再次与引理 1 矛盾。

如果一个级联协议 T 满足：对于某个 i，$lt(\overline{N_i(X,Y)}) \subseteq \{E_Y, D_Y, D_X\}$ 且对于某个 $j \geqslant 2$，$lt(\overline{N_j(X,Y)}) \subseteq \{E_x, D_x, D_Y\}$，则称 T 为**双重验证的**。

定理 2：每个双重验证的协议都是不安全的。

证明：令 T 为一个双重验证的协议，使得

且

（我们使用了缩写 $\overline{N}_i$ 表示 $N_i(X,Y)$。）写 $\overline{N}_1 = \alpha_1(X,Y)$，$\overline{N}_k = \gamma_k\alpha_1(X,Y)$ 和 $\overline{N}_l = \gamma_l\alpha_1(X,Y)$，其中 $r_j \in (\Sigma_2 \cup \Sigma_3)^*$。假设 T 是安全的。我们将导出一个矛盾。

根据定理 1，T 必须是平衡的。

情况 1，$E_{Y}\in lt(\alpha_{1}(X,Y))$：显然，(3) 要求 $\overline{\gamma}_l$ 应该包含 D_{Y} 但不包含 E_{Y}。这与引理 1 矛盾。

情况 2，$E_{X} \in \text{lt}(\alpha_{1}(X,Y))$ 且 $E_{Y} \notin \text{lt}(\alpha_{1}(X,Y))$：在这种情况下，(2) 要求 $\bar{\gamma}_{k}$ 包含 D_{X} 但不包含 E_{X}，与引理 1 矛盾。

定理 2 意味着在一个安全的级联协议 T 中，如果接收者 Y 能够解码加密消息 M，那么发送者 X 不能通过简单地解密发送回 X 的一些消息来获得 M。这意味着如果 X 在第一次传输后丢弃了 M，那么 X 应该无法重建 M。这个定理意味着示例 1 中的协议是不安全的（这是我们之前证明过的事实）。它还意味着 Diffie 和 Hellman [2] 中建议的协议（消息交换为 $E_{B}(D_{A}(M))$，$E_{A}(D_{B}(M))$）用于获得公钥认证是不安全的。

我们希望强调，我们的安全概念基于明文 M 是任意的这一假设。如果已知 M 的结构并且可以进行一致性检查，那么该协议不再被视为级联协议。在下一节中，我们将考虑一种可以使用消息的内部结构来实现安全性的情况。

三、名字戳协议

在第 I 节中，我们讨论了几种在加密前将名字附加到消息的协议。我们现在将引入一个包含此类协议的模型。

A. 非形式化描述

假设所有用户的姓名长度相同，例如 m 比特。对于任何字符串 $\gamma \in \{0,1\}^*$，我们将写 $\gamma = \text{head}(\gamma) \text{tail}(\gamma)$，其中 $\text{tail}(\gamma)$ 是 m 比特的后缀。一个用户 Y 可以对字符串 γ 应用以下任何操作：

a) 加密 E_{x}；
b) 解密 D_{Y}；
c) 附加 i_x；其中 $i_x \gamma = \gamma X$；
d) 名字匹配 d_x；其中 $d_x\gamma = \operatorname{head}(\gamma)$ 如果 $\operatorname{tail}(\gamma) = X$，否则未定义；
e) 删除 d，其中 $d\gamma = \operatorname{head}(\gamma)$。

名字 X 可以是任何用户的姓名，但 Y 唯一能应用的解密是 D_Y。以下等式显然成立。对于任何名字 X，

且

我们指出 $i_X d_X \neq 1$。

在名字戳协议下，用户传输的任何文本都是通过对最近接收到的文本应用一系列操作 a)–e) 获得的。特别是，当 d_x 应用于字符串 γ 时，除非 $\text{tail}(\gamma) = X$，否则传输不会继续。为确保通信完成，我们将要求任何两个正常用户 X, Y 之间传输的文本具有以下形式

使得在重复应用 (4) 后没有 d_A 保留。

和以前一样，允许破坏者拦截 X 和 Y 之间的所有文本，用操作 a)-e) 修改它们，并在任何对话中自由使用它们，无论是他发起的还是其他人发起的。通过这种方式，他可以获得大量字符串 $\gamma \in \{E_A, D_A, i_A, d_A, d\}$ 所有 $A\}^* M$。如果任何获得的 γ 可以通过重复使用 (4) 约简为 M，那么破坏者就成功地找到了 M。

B. 一些符号

考虑以下规则集：

对于任何字符串 $\gamma \in \{E_A, D_A, d_A, i_A, d\}$ 所有用户 $A\}^*$，令 $\bar{\gamma}$ 表示当规则 (5) 已被用于约简 γ 直到无法再进行替换时得到的字符串。显然，$\bar{\gamma}$ 是唯一的，与约简顺序无关。称 $\bar{\gamma}$ 为 γ 的**约简形式**。如果 $\bar{\gamma} = \gamma$，则字符串 γ 是**不可约的**。

C. 形式化模型

定义 6：一个两方名字戳协议 T 由一组字符串指定

其中 $F = \{z_{1}, z_{2}, \ldots, z_{9}\}$，$1 \leqslant i \leqslant t$，且 $1 \leqslant j \leqslant t'$（$t' = t$ 或 $t - 1$）。令 $\alpha_{i}(X,Y)$ 和 $\beta_{j}(X,Y)$ 表示当 $z_{1}, z_{2}, \dots, z_{9}$ 分别被 $D_{X}, D_{Y}, E_{X}, E_{Y}, i_{X}, i_{Y}, d_{X}, d_{Y}, d$ 替换时的字符串 $\tilde{\alpha}_{i}$ 和 $\tilde{\beta}_{j}$。令 $N_{1}(X,Y) = \alpha_{1}(X,Y)$，$N_{2}(X,Y) = \beta_{1}(X,Y)N_{1}(X,Y),\quad N_{3}(X,Y) = \alpha_{2}(X,Y)N_{2}(X,Y),\dots

,N_{2i}(X,Y) = \beta_{i}(X,Y)N_{2i - 1}(X,Y),$ $N_{2i + 1}(X,Y) = \alpha_{i + 1}(X,Y)N_{2i}(X,Y),\dots .$ 我们要求 $\overline{N_i(X,Y)}$ 不包含任何 d_A。

备注：$\{N_i(X,Y)M\}$ 是当 X 希望向 Y 发送明文 M 时，在 X 和 Y 之间传输的文本序列。$\overline{N_i(X,Y)}$ 不包含 d_A 意味着第 i 次传输是良定义的。

定义 7：令 X, Y, Z 为三个给定的不同用户。一个两方名字戳协议 T 是**不安全的**，如果存在一个字符串 $\gamma \in V_{Z,T}^{*}\{\overline{N_{i}(X,Y)}\}$ 使得 $\bar{\gamma} = \lambda$；集合 $V_{Z,T}$ 定义为

否则，T 是**安全的**。

备注：上述定义中 T 的安全性与 X, Y, Z 的选择显然无关。定义的动机与级联情况类似（见第 II-B 节），不再赘述。

D. 示例

1) 考虑示例 2 中给出的协议。在当前符号中，$\alpha_{1}(X,Y) = E_{Y}i_{X},\beta_{1}(X,Y) = E_{X}i_{Y}d_{X}D_{Y}$。我们还有 $\overline{N_1(X,Y)} = E_Yi_X$ 和 $\overline{N_2(X,Y)} = E_Xi_Y$。

2) 示例 3 中的协议对应于情况 $\alpha_{1}(X,Y) = E_{Y}i_{X}E_{Y}$，$\beta_{1}(X,Y) = E_{X}i_{Y}E_{X}D_{Y}d_{X}D_{Y}$。那么我们有 $\overline{N_1(X,Y)} = E_Yi_XE_Y$ 和 $\overline{N_2(X,Y)} = E_Xi_YE_X$。该协议是不安全的，因为字符串

满足 $\bar{\gamma} = \lambda$。（这个特定的 γ 实际上对应于示例 3 中破坏者使用的操作序列。）

E. 一个安全协议

我们现在证明示例 2 中的协议在我们的模型中是安全的。假设相反，存在 $\gamma \in V_{Z,T}^{*}\langle N_{i}(X,Y)\rangle$ 且 $\bar{\gamma} = \lambda$。我们将导出一个矛盾。

取这样的 $\gamma = v_{1}v_{2}\dots v_{l}\overline{N_{i}(X,Y)}$，其中 $v_{k}\in V_{Z,T}^{*}$ 的数量最少。假设 $i = 1$（另一种情况 $i = 2$ 可以类似处理）。从前一小节，我们有 $\overline{N_i(X,Y)} = E_Yi_X$ 和 $\overline{N_2(X,Y)} = E_Xi_Y$。由于 $\overline{\gamma} = \lambda$，$\gamma$ 中必须有一个 D_Y 来抵消 $N_{1}(X,Y)$ 中的 E_{y}。令 v_{j} 为包含此 D_Y 的字，那么 $v_{j} = \beta_{1}(W,Y) = E_{W}i_{Y}d_{W}D_{Y}$ 对于某个 W（因为 D_Y 只出现在 β_{1} 中）。这意味着 $j = l$，否则 $\gamma^{\prime} = v_{1}v_{2}\dots v_{j}N_{1}(X,Y)$ 将是一个比 γ 短的实例。现在有两种情况。

1) 如果 $W \neq X$，那么 $\overline{v_{l}N_{1}(X,Y)} = E_{W}i_{Y}d_{W}i_{X}$，且 $\overline{\gamma} = \overline{v_{1}v_{2}\cdots v_{l - 1}}E_{W}\underline{i_{Y}d_{W}i_{x}} \neq \lambda$。
2) 如果 $W = X$，那么 $v_{l}\overline{N_{1}(X,Y)} = E_{X}i_{Y} = \overline{N_{2}(X,Y)}$，因此字符串 $\gamma' = v_{1}v_{2}\dots v_{l - 1}\overline{N_{2}(X,Y)}$ 满足 $\overline{\gamma'} = \overline{\gamma}' = \lambda$，与 γ 的最小性矛盾。

证明完成。

F. 用于检查协议安全性的算法

我们将给出一个可以判定给定名字戳协议是否安全的算法。特别是，可以运行此算法来为上一节中的协议 1) 的安全性提供另一种证明。

给定一个由 $\{\alpha_i, \beta_j\}$ 指定的两方名字戳协议 T，我们将使用 n 表示输入长度 $\Sigma_i |\alpha_i| + \Sigma_j |\beta_j|$。本小节的其余部分致力于证明以下定理。

定理 3：存在一个算法，可以在 $O(n^{8})$ 时间内判定给定的两方名字戳协议 T 是否安全。

作为中间步骤，我们将证明定理 4，它本身也很有意义。原则上，破坏者 Z 可以启动与网络中任何用户的对话。下一个引理表明，我们可以假设 Z 只与 X 和 Y 对话。这种约简对于构建算法非常有用。让我们定义

引理 2：协议 T 是不安全的，当且仅当存在一个字符串 $\gamma \in S^{*}\{\overline{N_{i}(X,Y)}\}$ 使得 $\bar{\gamma} = \lambda$。

证明：只需证明，如果 T 是不安全的，则存在这样的 γ。在这种情况下，令 $\gamma' \in V_{Z,T}^*(\overline{N_i(X,Y)})$ 是一个满足 $\overline{\gamma'} = \lambda$ 的字符串。将 γ' 中所有当 $A \notin (X,Y,Z)$ 时的 E_A, i_A, d_A 替换为 E_Z, i_Z, d_Z，并令 γ 表示结果字符串。显然，$\overline{\gamma} = \lambda$。同时观察到 $\gamma \in S^*(\overline{N_i(X,Y)})$，因为如果 $A, B \notin (X,Y,Z)$，则 $\alpha_i(A,B)$ 和 $\beta_i(A,B)$ 变为 $\alpha_i(Z,Z)$ 和 $\beta_i(Z,Z) \in S$，否则为 $\alpha_i(A',B')$ 和 $\beta_i(A',B')$，其中 $A', B' \in (X,Y,Z)$，$A' \neq B'$。

定义 8：令 $\eta \in \{E_A, D_A, i_A, d|A = X, Y\}^*$ 为一个不可约字符串。用 $C(\eta)$ 表示所有满足 $\overline{\delta\eta} = \lambda$ 的不可约字符串 $\delta \in \{E_A, D_A, i_A, d_A, d|\}$ 所有 $A\}^*$ 的集合。

引理 3：如果 η 包含任何 d，则 $C(\eta) = \emptyset$。否则，令 $\eta = b_{1}, b_{2}, \dots, b_{t}$，那么 $C(\eta)$ 由所有字符串 $b_{t}^{c}b_{t-1}^{c} \dots b_{1}^{c}$ 组成，其中 $(E_{A})^{c} = D_{A}$，$(D_{A})^{c} = E_{A}$，$(i_{A})^{c} = d_{A}$ 或 d。

证明：这源于 d 没有左逆，以及 b_{i}^{c} 是仅有的满足 $\overline{b_{i}^{c}b_{i}} = \lambda$ 的不可约字符串这一事实。

记 $\rho_{k} = \overline{N_{k}(X,Y)}$，并令 $\rho_{i_1}, \rho_{i_2}, \dots, \rho_{i_s}$ 为那些不包含 d 的 ρ_{k}。

引理 4：协议 T 是不安全的，当且仅当存在一个字符串 $\gamma \in S^{*}$ 使得对于某个 $1 \leqslant j \leqslant s$，$\overline{\gamma} \in C(\rho_{i_j})$。

证明：

充分性：如果 $\gamma \in S^{*}$ 且 $\overline{\gamma} \in C(\rho_{i_j})$，那么 $\gamma' = \gamma \rho_{i_j} \in S^{*}\langle \overline{N_i(X,Y)} \rangle$ 且 $\overline{\gamma'} = \lambda$。因此根据引理 2，$T$ 是不安全的。

必要性：如果 T 是不安全的，那么根据引理 2，存在一个字符串 $\gamma' = \underline{\gamma N_k(X,Y)}$ 其中 $\gamma \in S^*$ 且 $\overline{\gamma'} = \lambda$。这意味着 $\overline{\gamma} \rho_k = \lambda$，因此根据引理 3，对于某个 j，有 $\overline{\gamma} \in C(\rho_{i_j})$。

我们将证明以下命题。

命题 1：给定一组字符串 $S = \{h_1, h_2, \dots, h_p\}$ 和一个字符串 ρ，其中

且

可以在 $O(q^7)$ 时间内判定是否存在一个字符串 $\gamma \in S^*$ 使得 $\bar{\gamma} \in C(\rho)$。（q 定义为 $\sum_{i=1}^{p} |h_i| + |\rho|$。）

命题 1 通过以下论证蕴含定理 3。给定一个由 $\{\alpha_i, \beta_i\}$ 指定的协议 T，我们首先计算所有 i 的 $N_i(X, Y)$ 然后 $\rho_i = \overline{N_i(X, Y)}$，时间为 $O(n^2)$。（注意每个 $N_i(X, Y)$ 的长度最多为 $O(n)$。）考虑那些不包含 d 的 ρ_i。对于每个这样的 ρ_i，使用命题 1 判定是否存在 $\gamma \in S^*$ 使得 $\overline{\gamma} \in C(\rho_i)$，其中 S 由 (6) 给出。那么根据引理 4，协议是不安全的，当且仅当对于某个 ρ_i 存在这样的 γ。总时间是

命题 1 仍有待证明。我们将考虑一个更一般的设置。

G. 扩展字问题

令 $\Sigma = \{a_{1}, a_{2}, \dots, a_{r}\}$ 为一个字母表，即一组不同的符号。我们称 $u \to v$ 为一条**转换规则**，其中 $u \in \Sigma^{+}$ 且 $v \in \Sigma^{*}$。令 $\Gamma = \{u_{1} \to v_{1}, u_{2} \to v_{2}, \dots, u_{q} \to v_{q}\}$ 为一组转换规则。对于两个字符串 $\gamma, \delta \in \Sigma^{*}$，如果 γ 可以通过重复使用 Γ 中的规则（即用 v_{i} 替换子串 u_{i}）转换为 δ，我们将记为 $\gamma \mapsto_{\Gamma} \delta$。对于一个子集串 $\eta = G_{1}, G_{2}, \dots, G_{q}$，其中 G_{i} 是 Σ 的子集，令 $L(\eta) = \{\gamma \mid \gamma = g_{1}, g_{2}, \dots, g_{q}\}$，其中 $g_{i} \in G_{i}$。如果对于某个 $\rho \in L(\eta)$，有 $\gamma \mapsto_{\Gamma} \rho$，我们将使用记号 $\gamma \mapsto_{\Gamma} L(\eta)$。$(\Sigma, \Gamma)$ 的**扩展字问题**可以表述如下。

给定一组输入字符串 $\delta_1, \delta_2, \dots, \delta_p (\delta_i \in \Sigma^*)$ 和一个子集串 $\eta = G_1, G_2, \dots, G_q (G_i \subseteq \Sigma; G_i \neq \emptyset)$，确定是否存在一个连接 $\Delta = \delta_{i_1}, \delta_{i_2}, \dots, \delta_{i_s}$ 使得 $\Delta \mapsto_{\Gamma} L(\eta)$。

备注：输入长度 n 定义为 $\Sigma_{i}|\delta_{i}| + \Sigma_{j}|G_{j}|$。

一般来说，扩展字问题是不可判定的，因为它包含了 0 型语言的成员资格问题作为特例，而后者已知是不可判定的（例如，见 Hopcroft 和 Ullman [5]）。然而，我们将证明对于一类特殊的输入，该问题可以在多项式时间内求解。

定义 9：形式为 $a_{i}a_{j}\rightarrow \lambda$ 的转换规则称为**抵消规则**。

定理 4：令 Σ 为一个字母表，Γ 为一组抵消规则。那么 (Σ, Γ) 的扩展字问题可以在 $O(n^{7})$ 时间内求解，其中 n 是输入长度。

定理 4 通过以下论证蕴含命题 1。令

命题 1 中所述的问题，输入为 $h_1, h_2, \dots, h_p, \rho$，可以作为 (Σ, Γ) 的扩展字问题来求解。输入是 $\delta_1, \delta_2, \dots, \delta_p$ 和一个子集串 $\eta = G_1, G_2, \dots, G_q$，其中 $\delta_i = h_i$ 且 η 使得 $L(\eta) = C(\rho)$。输入长度是线性相关的。因此证明定理 4 将完成定理 3 的证明。

为了准备定理 4 的证明，我们定义几个术语。令 $\delta_1, \delta_2, \dots, \delta_p$ 为 Σ^* 中的输入字，令 $\eta = G_1, G_2, \dots, G_q$ 为输入的子集序列（$G_i \subseteq \Sigma$）。不失一般性，我们可以假设所有 i 都有 $\delta_i \neq \lambda$。用 I, I' 表示 $\delta_1, \delta_2, \dots, \delta_n$ 的所有真前缀和后缀的集合（包括 λ，但不包括 δ_i）。令 J 为 η 的所有子串的集合，J_l 为 η 中所有长度为 l 的子串的集合。对于每个 $w \in J$，令

使得 $geb\Rightarrow_{\Gamma}L(\omega)\rangle$

我们强调每个 $w \in J$ 都具有 $G_{i}, G_{i+1}, \dots, G_{j}$ 的形式，其中 $G_{k} \subseteq \Sigma$。

引理 5：R_{λ} 可以在 $O(n^{7})$ 时间内计算。

证明见附录 II。

定理 4 的证明：我们通过"动态规划"对 l 进行归纳来计算 R_{w}（$w \in J_{l}$）。首先，我们计算 R_{λ}。现在令 $l > 0$，并假设 R_{w} 已经对所有 $w \in J_{0} \cup J_{1} \cup \dots \cup J_{l-1}$ 计算完毕。对于每个 $w \in J_{l}$，我们将计算 R_{w}。令 $w = G_{j}u$。对于每个 $g \in I'$，$b \in I$，让我们判断是否 $(g, b) \in R_{w}$。假设 $g\delta_{i_1}\delta_{i_2}\dots \delta_{i_s}b \Rightarrow_{\Gamma} L(G_ju)$。由于抵消规则不会创建新符号，$g\delta_{i_1}\delta_{i_2}\dots \delta_{i_s}b$ 必须具有 $\rho a_k\rho'$ 的形式，对于某个 $a_k \in G_j$，其中 $\rho \Rightarrow_{\Gamma}\lambda$ 且 $\rho' \Rightarrow_{\Gamma}L(u)$。为了覆盖 ρ 和 ρ' 的所有可能断点，我们采用以下过程：

1) 如果 $g = a_{k}g_{1}$ 且 $a_{k}\in G_{j}$，判断是否 $(g_1,b)\in R_u$。
2) 对于每个 δ_j 和 $a_k \in G_j$ 在 δ_j 中的每次出现，写 $\delta_j = s a_k s'$。判断是否 $(g, s) \in R_\lambda$ 和 $(s', b) \in R_u$ 都成立。
3) 如果 $b = b_{1}b_{2}$ 且 $b_{2} \in L(w)$，判断是否 $(g, b_{1}) \in R_{\lambda}$。

如果上述任何测试得到"是"的答案，则设置 $(g, b) \in R_w$；否则 $(g, b) \notin R_w$。很容易检查上述过程是否正确判断了 $(g, b) \in R_w$ 与否。为了找出运行时间，注意每个三元组 (w, g, b) 最多需要时间

因此，计算所有 $w \in J_{l}$ 的 R_{w} 所需的时间是

因此，对于 $l = 1,2,\dots ,|\eta |$ 的总计算时间是 $O(n^{7})$。由于扩展字问题有解当且仅当 $(\lambda ,\lambda)\in R_{\eta}$，定理 4 的证明完成。

四、不耐烦的破坏者

为了攻破如前几节所定义的不安全协议，破坏者可能需要成为对话的接收者。在本节中，我们关注可以被不耐烦的破坏者（即仅发起对话而不依赖他人对他说话的人）攻破的协议的特征描述（或判定过程）。

对于名字戳协议，这对应于安全性定义（定义 7）的修改。也就是说，应该从 $V_{Z,T}$ 的定义中省略 $\{\alpha_{j}(A,B)\}$ 项（见 (6)）。

定理 5：存在一个算法，可以在 $O(n^{8})$ 时间内判定给定的两方名字戳协议 T 是否对不耐烦的破坏者是安全的。

证明：证明与定理 3 的证明相同，只是应该从 (7) 中省略 $\langle \alpha_{i}(A,B)\rangle$ 项。

对于级联协议，安全性定义（定义 3）应修改如下。如果存在某个 $\gamma \in (\Sigma_1(Z) \cup \Sigma_3)^*$ 使得对于某个 $N_i(X,Y)$，有 $\overline{\gamma N_i(X,Y)} = \lambda$，则 T 是**不安全的**（对不耐烦的破坏者而言）；否则 T 是**安全的**。我们可以获得一个类似于定理 1 的特征描述。

定理 6：令 X, Y 为不同的用户名。一个两方级联协议 $T = \{\tilde{\alpha}_i, \tilde{\beta}_j\}$ 对不耐烦的破坏者是安全的，当且仅当对于每个 $k \geqslant 1$，

1) $lt(\overline{N_k(X,Y)})\cap \langle E_X,E_Y\rangle \neq \emptyset$；
2) $\beta_{k}(X,Y)$ 关于 Y 具有平衡性质。

尽管这个结果的陈述很简单，但证明相当复杂。本节的其余部分致力于证明定理 6。在下文中，字符串总是指 E 和 D 的字符串。令 A 为任何用户名。

定义 10：令 η 为一个字符串。如果对于某个 $X, Y \neq A$，以下任一成立，则 η 的子串 π 称为一个 **A-子串**：

1) $\eta = \eta_{1}D_{x}\pi D_{Y}\eta_{2}$；
2) $\eta = \eta_{1}D_{x}\pi$；
3) $\eta = \pi D_Y\eta_2$。

定义 11：如果 η 的每个 A-子串 π 关于 A 都具有平衡性质，则称字符串 η 是**强 A 平衡的**。

引理 6：令 η 为强 A 平衡的字符串。
1) 如果 $\eta = \eta_1D_B\eta_2$ 且 $B\neq A$，那么 η_{1} 和 η_{2} 都是强 A 平衡的。
2) 如果 $\eta = \eta_1\eta_2$ 且 $E_A \notin lt(\eta_2)$，那么 η_1 是强 A 平衡的。

证明：容易看出，η 是强 A 平衡的当且仅当每个不包含任何 D_B（$B \neq A$）的 A 子串关于 A 都具有平衡性质。这蕴含 1)。

关于 A 的平衡性质关注的是当 D_A 出现时 E_A 的出现。因此，通过移除不包含 E_A 的后缀或前缀，我们不能改变平衡性质或强平衡性质。这证明了 2)。

定理 6 证明中的关键思想是以下引理中呈现的性质。

引理 7：令 γ, δ 为任何以约简形式给出的强 A 平衡字符串。如果

那么 $\overline{\gamma\delta} \neq \lambda$。

证明：我们通过对 n（字 $\gamma \delta$ 中 D_{A} 的数量）进行归纳来证明引理。

当 $n = 0$ 时，引理显然成立。现在假设 $\gamma \delta$ 中出现了一个 D_A。我们将假设它在 γ 中，并且 δ 不包含 D_A。（D_A 在 δ 中的情况可以类似处理。）根据假设，字符串 γ, δ 是约简形式。因此，$\overline{\gamma \delta} = \lambda$ 意味着 γ 不包含任何 E_A，这与 γ 的平衡性质矛盾。因此引理对 $n = 1$ 成立。

对于归纳步骤，令 $n > 1$。假设引理对于每个 γ', δ' 成立，只要 $\gamma' \delta'$ 包含最多 $n - 1$ 个 D_A。令 γ, δ 使得 $\gamma \delta$ 包含 n 个 D_A，并且 γ, δ 满足归纳假设。我们希望证明 $\overline{\gamma \delta} \neq \lambda$。

我们通过反证法证明。假设 $\overline{\gamma}\delta = \lambda$。根据假设 γ 和 δ 是约简形式；因此 $\delta = \gamma^c$。这意味着 γ 和 δ 包含相同数量的来自集合 $\{E_A, D_A\}$ 的操作符。让我们假设 $\gamma = \gamma_2D_A^+\gamma_1$，其中 $lt(\gamma_1)\cap \{E_A,D_A\} = \emptyset$。（$\gamma = \gamma_{2}E_{A}^{+}\gamma_{1}$ 的情况类似。）在这种情况下，$\delta = \delta_{1}E_{A}^{+}\delta_{2}$，其中 $\delta_{1} = \gamma_{1}^{c}$ 且 $\delta_{2} = \gamma_{2}^{c}$。

字符串 γ 是强 A 平衡的。因此，γ_{2} 应该具有 $\gamma_{3}E_{X}$ 的形式，对于某个 $X \neq A$。（否则，$\gamma = \gamma_{3}D_{X}D_{A}^{+}\gamma_{1}$ 对于 $X \neq A$ 且 $E_{A} \notin lt(\gamma_{1})$，这与 $D_{A}^{+}\gamma_{1}$ 关于 A 具有平衡性质这一事实矛盾。）这意味着 $\delta_{2} = D_{x}\delta_{3}$ 且 $\delta_{3} = \gamma_{3}^{c}$。

根据引理 6 以及 $lt(\gamma_1) \cap \{E_A, D_A\} = \emptyset$ 这一事实，我们得出 γ_3 和 δ_3 是强 A 平衡的。此外，$\gamma = \gamma_3 E_x D_A^+ \gamma_1$ 且 $E_A \notin lt(\gamma_1)$ 这一事实意味着 $E_A \in lt(\gamma_3)$，这又意味着 $D_A \in lt(\delta_3)$。归纳假设意味着 $\overline{\gamma_3 \delta_3} \neq \lambda$，这与我们假设 $\overline{\gamma \delta} = \lambda$ 矛盾。

引理 8：令 $\Sigma_Y = \{\beta_i(X,Y)\}$ 对于所有 i 和所有用户 X。如果 Σ_Y 的每个成员关于 Y 都具有平衡性质，那么对于每个字符串

η 是强 Y 平衡的。

证明：证明与附录 I 中引理 1 的证明非常相似。强平衡性质是具有链接性质（附录 I）的一个特例。证明可以按照相同的思路进行，在目前的情况下只关注一方 Y。

我们现在准备证明定理 6。

定理 6 的证明：必要性部分与定理 1 完全相同。

充分性：假设相反，存在 $P \in (\Sigma_1(Z) \cup \Sigma_3)^*$ 使得对于某个 k

（我们将使用缩写 N_{k} 表示 $N_{k}(X,Y)$。）

情况 A，如果 $E_{Y} \notin \text{lt}(\overline{N}_{k})$：性质 2) 和引理 8 意味着字符串 $\overline{N}_{k}$ 是强 Y 平衡的。因此，$\overline{N}_{k}$ 不能包含任何 D_{Y}。于是我们有 $\overline{N}_{k} = E_{X}^{+}$，否则 1) 将不成立。这意味着 $\overline{P} = D_{X}^{+}$。现在，条件 2) 说明对于每个 A，$\beta_{j}(A,X)$ 关于 X 具有平衡性质。因此，根据引理 8，$\overline{P}$ 是强 X 平衡的，这与 $\overline{P} = D_{X}^{+}$ 这一事实矛盾。

情况 B，$E_{Y} \in \text{lt}(\overline{N}_{k})$：在这种情况下，根据引理 8，$\overline{N}_{k}$ 和 $\overline{P}$ 是强 Y 平衡的。然而，引理 7 意味着

这提供了所需的矛盾。

附录 I 引理 1 的证明

令 Z 为一个特殊的用户名。我们将探索 $(\Sigma_1 \cup \Sigma_2 \cup \Sigma_3)^*$ 中字符串的结构，从中我们将导出引理 1。（在本附录中，我们自始至终用 Σ_1 表示 $\Sigma_1(Z)$。）

定义：令 π 为一个字符串，A 为一个用户名。如果满足以下条件，则称 π 是 **A 平衡的**：$\pi = D_X \delta D_Y$，其中 $X, Y \neq A$ 且 $lt(\delta|_A) \cap D \subseteq \{D_A\}$ 蕴含 $\delta|_A$ 关于 A 具有平衡性质。

如果一个字符串 η 的每个子串 π 对于所有 $A \neq Z$ 都是 A 平衡的，则称 η 具有**链接性质**。

引理 9：令 T 为一个平衡级联协议。令 μ 为任何具有链接性质的字符串。对于来自 Σ_1^+ 或 Σ_2 或 Σ_3 的任何字符串 η，$\mu \eta$ 和 $\eta \mu$ 满足链接性质。

证明：只需证明 $\mu \eta$ 具有链接性质；另一种情况由对称性可得。令 $A \neq Z$ 为网络的任何用户。我们必须证明 $D_Z \mu \eta D_Z$ 的每个子串都是 A 平衡的。

首先考虑 η 不包含 D_{A} 的情况。在这种情况下，$(\mu \eta)|_{A}$ 中 D_{A} 的数量不会增加，因此，$D_{Z}\mu \eta D_{Z}$ 的每个子串都是 A 平衡的，因为 $D_{Z}\mu D_{Z}$ 的每个子串都是 A 平衡的。

接下来，假设 η 包含 D_A。在这种情况下，η 应该是某个 $\alpha_k(A,B)$ 或 $\beta_k(B,A)$，对于某些用户 A 和 B。因此 η 关于 A 具有平衡性质。此外，η 不包含任何 D_{u}（$u \neq A$）。令 $\pi = D_{X}\delta D_{Y}$，其中 $X, Y \neq A$，是 $D_{Z}\mu \eta D_{Z}$ 的一个子串。如果 $D_{Y} \in lt(\mu)$，那么 π 是 A 平衡的，因为 μ 满足链接性质。否则，D_{Y} 应该是最右边的 D_{Z}，η 不能包含 D_{Y}（因为 $Y \neq A$）。

在这种情况下，$\pi = D_x\delta 'eta D_Z$，其中 $\delta = \delta^{\prime}\eta$，因为 D_{x} 也不能在 η 中。我们必须证明如果 $\delta |_{A}$ 包含 D_{A}，那么它必须包含 E_{A}。假设相反，$\delta |_{A}$ 包含 D_{A} 但不包含 E_{A}。可以证明，要么 η 要么 $\delta^{\prime}|_{A}$ 应该包含 D_{A} 而不包含 E_{A}，因为在 $\delta |_{A}$ 中最多可以抵消一个 D_A^+ 或 E_A^+ 块（因为只能在其中进行 A 约简）。这导致与 η 具有平衡性质的假设或 μ 具有链接性质的假设相矛盾。

引理 10：令 T 为一个平衡级联协议。那么 $(\Sigma_1 \cup \Sigma_2 \cup \Sigma_3)^*$ 中的每个字符串 η 都具有链接性质。

证明：对于每个 $\eta \in (\Sigma_1 \cup \Sigma_2 \cup \Sigma_3)^*$，写 $\eta = w_1 \cdots w_n$，其中每个 w_i 要么在 Σ_1^+ 中，要么在 Σ_2 中，要么在 Σ_3 中。可以通过对 n（η 中字的数量）进行简单归纳，使用引理 9 来完成证明。

引理 11：对于任何字符串 η 和任何 Z，如果 η 具有链接性质，那么 $\bar{\eta}$ 也具有相同的性质。

证明：只需证明，如果一个字符串具有该性质，那么通过约简任何对 $D_X E_X$ 或 $E_X D_X$（对于任何 X）得到的新字符串也具有该性质。令 η 为任何具有链接性质的字符串。假设

令 A 为任何不同于 Z 的用户。我们必须证明 $D_Z \eta_1 \eta_2 D_Z$ 的每个子串都是 A 平衡的。

如果 $A = X$，那么 $D_Z \eta_1 \eta_2 D_Z$ 的每个子串都是 A 平衡的，因为 $D_Z \eta D_Z$ 的对应子串是 A 平衡的。在下文中，我们假设 $A \neq X$。

令 D_{Y} 为 $D_{Z}\eta_{1}$ 中从右起第一个非 D_{A} 的 D_{V}，令 D_{W} 为 $\eta_{2}D_{Z}$ 中从左起第一个这样的 $D_{V}(V \neq A)$。那么我们可以写

η 具有链接性质这一事实意味着 $D_{Y}\delta_{1}E_{X}D_{X}$ 和 $D_{X}\delta_{2}D_{W}$ 是 A 平衡的。那么很容易看出 $D_{Y}\delta_{1}\delta_{2}D_{W}$ 也是 A 平衡的。证明完成。

注意引理反过来不成立，即如果 $\bar{\eta}$ 具有链接性质，并不蕴含 η 也具有该性质。

示例：令 $\eta = E_W D_X E_X E_Y D_W D_Y E_Y$，那么 $\bar{\eta}$ 具有链接性质但 η 不具有。

引理 1 的证明：我们必须证明对于每个字符串 $\eta \in (\Sigma_1 \cup \Sigma_2 \cup \Sigma_3)^*$，约简字符串 $\bar{\eta}$ 关于每个 $A \neq Z$ 具有平衡性质。

假设相反，对于某个 A，$\overline{\eta}$ 关于 A 不具有平衡性质。这意味着 $\overline{\eta}$ 包含一个 D_A 但不包含任何 E_A。这表明 $\overline{\eta}$ 不具有链接性质。然而，根据引理 10 和 11，$\overline{\eta}$ 必须具有链接性质。这导致矛盾。

附录 II 引理 5 的证明

本附录的目的是证明 R_{λ} 可以在 $O(n^{7})$ 时间内计算。（符号见第 III 节。）对于

使得 $g\delta_{i_1}\delta_{i_2}\dots\delta_{i_j}b\mapsto_{\Gamma}\lambda\rangle$

显然，

定义 $V_{0} = Q_{0}$。我们将给出一个过程 K，它在给定 V_{l-1} 后生成一个集合 $V_{l} \subseteq I' \times I$。考虑通过此过程迭代生成的序列 $V_{0}, V_{1}, V_{2}, \dots$。我们将证明序列 $\{V_{l}\}$ 满足以下性质：

P1) 对所有 $l \geqslant 0$，$V_{l} \subseteq V_{l+1}$ 且 $V_{l} \subseteq R_{\lambda}$；
P2) 对所有 $l \geqslant 0$，$Q_{l} \subseteq V_{l}$；
P3) 对所有 $l \geqslant |I| \cdot |I'|$，$V_{l+1} = V_l$。

根据 (8) 和 P1-P3)，可以得出

因此，如果我们首先计算 $V_{0} = Q_{0}$，然后应用 $|I| \cdot |I^{\prime}|$ 次过程 K，我们将得到所需的 R_{λ}。计算 V_{0} 所需的时间很容易看出是 $O(n|I| \cdot |I'|)$。令 $\mathrm{cost}(K)$ 表示过程 K 的最大运行时间，那么计算 R_{λ} 的总时间是 $O((n + \mathrm{cost}(K))|I| \cdot |I'|) = O(n^{3} + n^{2} \mathrm{cost}(K))$。我们将证明 $\mathrm{cost}(K) = O(n^{5})$，从而得到 $O(n^{7})$ 的总运行时间。

我们现在描述过程 K。假设 V_{l-1} 以维度为 $|I'| \times |I|$ 的矩阵表示形式给出，我们将描述如何生成 V_{l}。

过程 **K**

我们按 $s = |g| + |b|$ 长度的递增顺序，逐个处理 $I' \times I$ 中的对 (g, b)。对于每个 (g, b)，如果它在 V_{l-1} 中，我们将其包含在 V_{l} 中，否则我们根据情况执行以下步骤：

情况 1，$|g| = 0$，$|b| = 0$：
步骤 a) 对于每个 $1 \leqslant k, j \leqslant p$，测试是否 $(a_{i}a_{r} \to \lambda) \in \Gamma$ 且 $(\delta_{k}', \delta_{j}') \in V_{l-1}$，其中 $\delta_{k} = a_{i}\delta_{k}'$，$\delta_{j} = \delta_{j}'a_{r}$；如果答案为"是"，则令 $(g, b) \in V_{l}$。
步骤 b) 对于每个 $1 \leqslant j \leqslant p$ 和每个划分 $\delta_j = st$（(s, t) 可能为 λ），测试是否 $(g, s) \in V_{l-1}$ 和 $(t, b) \in V_{l-1}$；如果答案为"是"，则令 $(g, b) \in V_l$。

情况 2，$|g| = 0$，$b = b _ {1} a _ {j}$：
步骤 a) 对于每个 $1 \leqslant k \leqslant p$，测试是否 $(a_{i}a_{j} \to \lambda) \in \Gamma$ 且 $(\delta_k', b_1) \in V_{l-1}$，其中 $\delta_k = a_i\delta_k'$；如果答案为"是"，则令 $(g, b) \in V_l$。
步骤 b) 对于每个 $1 \leqslant k \leqslant p$ 和每个划分 $\delta_k = st$（(s, t) 可能为 λ），测试是否 $(g, s) \in V_{l-1}$ 和 $(t, b) \in V_{l-1}$；如果答案为"是"，则令 $(g, b) \in V_l$。

情况 3，$g = a _ {k} g _ {1}$，$| b | = 0$：
步骤 a) 对于每个 $1 \leqslant j \leqslant p$，测试是否 $(a_k a_i \to \lambda) \in \Gamma$ 且 $(g_1, \delta_j') \in V_{l-1}$，其中 $\delta_j = \delta_j' a_i$；如果答案为"是"，则令 $(g, b) \in V_l$。
步骤 b) 对于每个 $1 \leqslant k \leqslant p$ 和每个划分 $\delta_k = st$，测试是否 $(g, s) \in V_{l-1}$ 和 $(t, b) \in V_{l-1}$；如果答案为"是"，则令 $(g, b) \in V_l$。

情况 4，$g = a _ {k} g _ {1}$，$b = b _ {1} a _ {j}$：
步骤 a) 测试是否 $(a_{k}a_{j}\to \lambda)\in \Gamma$ 且 $(g_{1},b_{1})\in V_{l}$；如果答案为"是"，则令 $(g,b)\in V_l$。
步骤 b) 对于每个 $1 \leqslant k \leqslant p$ 和每个划分 $\delta_k = st$（(s, t) 可能为 λ），测试是否 $(g, s) \in V_{l-1}$ 和 $(t, b) \in V_{l-1}$；如果答案为"是"，则令 $(g, b) \in V_l$。

注释：如果 (g, b) 在这些步骤之后未被包含在 V_{l} 中，那么 $(g, b) \notin V_{l}$。

结束过程 K

很容易检查，对于每一对 (g, b)，处理所需的时间最多为 $O(n|I| \cdot |I'|) = O(n^3)$。因此 $\mathrm{cost}(K) = O(n^3|I| \cdot |I'|) = O(n^5)$。

我们现在可以通过证明序列 $\langle V_l \rangle$ 满足 P1)-P3) 来完成引理的证明。我们观察到，每当在过程 K 中将一个 (g, b) 添加到 V_l 时，条件自然地给出了一个可以约简为 λ 的字符串 $\gamma \in g(\delta_i)^*b$ 的构造。我们省略了一个直接的证明。这就确立了 P1)。

为了证明 P3)，注意 V_{l} 从 V_{l-1} 的构造不显式依赖于 l。因此，一旦我们发现 $V_{l} = V_{l-1}$，那么 $V_{l} = V_{l+1} = V_{l+2} = \cdots$。这个条件必须在某个 $l \leqslant |I| \cdot |I'|$ 时达到，因为任何 V_{i} 中最多有 $|I| \cdot |I'|$ 个元素。这证明了 P3)。

我们通过对 l 的归纳来证明 P2)。情况 $l = 0$ 是平凡的，因为 $V_0 = Q_0$。现在假设我们已经对小于 l 的所有值证明了 P2)，我们将证明 P2) 对 $l (l > 0)$ 成立。

我们需要证明，对于任何 $(g,b)\in Q_l$，必须有 $(g,b)\in V_l$。我们通过对值 $s = |g| + |b|$ 进行归纳来证明这一点。对于任何 $s\geqslant 0$，让我们假设该陈述对所有满足 $|g| + |b| < s$ 的 (g,b) 成立；我们将证明当 $|g| + |b| = s$ 时的陈述。有四种情况需要考虑，取决于是否 $|g| = 0$ 和是否 $|b| = 0$。我们将考虑 $|g| > 0$ 且 $|b| > 0$ 的情况，并将其他三种情况留作练习。

如果 $(g, b) \in Q_{l-1}$，那么根据归纳假设，$(g, b) \in V_{l-1} \subseteq V_l$。因此我们可以假设 $(g, b) \in Q_l - Q_{l-1}$。写 $g = a_k g_1$，$b = b_1 a_j$，并假设

在一个约简到 λ 的过程中，要么最后一步是 $a_{k}a_{j}\rightarrow \lambda$，要么最左边的 a_{k} 的抵消是与某个 δ_{i_u} 中的符号 a_{v}。在前一种情况下，$(g_1,b_1)\in Q_l$，并且由于 $|g_{1}| + |b_{1}| < s$，根据归纳假设，我们有 $(g_{1},b_{1})\in V_{l}$；这意味着 (g,b) 将在过程 K 的执行过程中被包含在 V_{l} 中（情况 4，步骤 a）。在后一种情况下，存在一个划分 $\delta_{i_u} = st$ 使得 $(g,s)\in Q_{l - 1}$ 且 $(t,b)\in Q_{l - 1}$；这意味着 (g,b) 在过程中被添加到 V_{l}（情况 4，步骤 b）。这完成了归纳。

我们已经证明了 P2)，因此证明了引理 5。

参考文献

[1] W. Diffie 与 M. Hellman, "密码学的新方向", IEEE 信息论汇刊，卷 IT-22，第 644–654 页，1976年。

[2] W. Diffie 与 M. Hellman, "多用户密码技术", 1976年 AFIPS 全国计算机会议论文集。蒙瓦勒，新泽西州，AFIPS 出版社，第 109-112 页。

[3] D. Dolev, "拜占庭将军再次出击", 算法杂志，卷 3，第 14-30 页，1982年。

[4] M. A. Harrison, W. L. Ruzzo, 与 J. D. Ullman, "操作系统中的保护", ACM 通讯，卷 19，第 461-471 页，1976年。

[5] J. E. Hopcroft 与 J. D. Ullman, 形式语言及其与自动机的关系。马萨诸塞州雷丁：Addison-Wesley，1969年。

[6] R. Lipton 与 L. Snyder, "判定科目安全性的线性时间算法", ACM 杂志，卷 24，第 455-464 页，1977年。

[7] R. C. Merkle, "公钥密码学协议", BNR 技术报告。帕洛阿尔托，加利福尼亚州，1980年。

[8] R. M. Needham 与 M. D. Schroeder, "在大型计算机网络中使用加密进行认证", ACM 通讯，卷 2，第 993–999 页，1978年。

[9] M. Pease, R. Shostak, 与 L. Lamport, "在存在故障的情况下达成协议", ACM 杂志，卷 27，第 228-234 页，1980年。

[10] G. J. Popek 与 C. S. Kline, "计算机网络中的加密协议、公钥算法和数字签名", 安全计算基础，R. A. Demillo 等编辑。纽约：学术出版社，1978年。

[11] R. L. Rivest, A. Shamir, 与 L. Adleman, "一种获取数字签名和公钥密码系统的方法", ACM 通讯，卷 21，第 120–126 页，1978年。

密钥保护的模块化方法

CHARLES ASMUTH 与 JOHN BLOOM

摘要——本文提出了一种密钥保护方案（阈值方案）的方法，其中影子（shadows）是与原始密钥相关联的一个数的同余类。该方案的一个变体提供了有效的错误检测，甚至可以暴露故意篡改。该方案与 Shamir 的插值法某些潜在的相似性，使得可以将这些保护特性也融入到该插值法中。

一、引言

我们考虑以下问题。给定一个密钥 x，希望将其分解为影子 $y_{1}, \dots, y_{n}$，使得密钥 x 可以从 y_{i} 中的任意 r 个恢复，但本质上从 s 个或更少的任何 y_{i} 中无法推导出任何信息。（见 [1]，亦见 [4]。）我们将任何实现此目标的方法称为"密钥保护方案"。此类方案也称为阈值方案，并具有密钥保护以外的用途。

此类方案的价值取决于许多特性。其中一些是

1) 密钥分解和恢复的效率，
2) 该方法对随机错误或故意篡改的敏感性，
3) r, s 和 n 之间的关系。

最好能有 $r = s + 1$。这是最尖锐的安排。然而，人们可能会考虑

稿件收到日期：1981年3月10日；修订日期：1982年7月20日。本文曾在1980年12月3日于德克萨斯州休斯顿举行的全国电信会议上发表。

C. Asmuth 就职于德克萨斯农工大学数学系，大学城，德克萨斯州 77843。

J. Bloom 就职于雪佛龙油田研究公司，拉哈布拉，加利福尼亚州。

牺牲一些这种尖锐性，如果在某些其他特性（例如速度）上有补偿性的改进。

Shamir 的多项式插值法 [4] 是具有最大尖锐性的方法之一。选择某域中的一组数 $\{x_0, x_1, \dots, x_n\}$。构造一个次数为 $r - 1$ 的多项式 P，使得 $P(x_0) = x$。数 $y_i = P(x_i)$（$i = 1$ 到 n）是影子。密钥通过在 x_0 处评估拉格朗日插值多项式来恢复。正如我们将看到的，这种方法对错误有些敏感。此外，通过通常的插值公式恢复密钥需要 $O(r\log^2 r)$ 次运算。本文的模方法仅需要 $O(r)$ 次运算。它也具有最大的尖锐性。此外，它很容易修改以包含在恢复密钥之前检查影子有效性的选项。

二、基本方法

选择一组整数 $\{p, m_1 < m_2 < \dots < m_n\}$，满足以下条件：

1) $(m_i, m_j) = 1$ 对于 $i \neq j$，
2) $(p, m_i) = 1$ 对于所有 i，
3) $\prod_{i=1}^{r} m_{i} > p \prod_{i=1}^{r-1} m_{n-i+1}$。

这里，如前所述，n 表示影子的数量。任何 r 个影子足以恢复密钥。对素数密度的估计表明，可以很容易地找到素数 m_i 来满足 3)。找到合数 m_i 更容易。最后，令 $M = \prod_{i=1}^{r} m_i$。

分解过程始于密钥 x；我们假设 $0 \leqslant x < p$。令 $y = x + Ap$，其中 A 是一个任意整数，满足条件 $0 \leqslant y < M$。然后令 $y_i \equiv y (\bmod m_i)$ 为影子。

专业术语中英文对照表

英文术语	中文翻译
Public key encryption	公钥加密
Protocol	协议
Saboteur	破坏者
Passive eavesdropper	被动窃听者
Active saboteur	主动破坏者
Cascade protocols	级联协议
Name-stamp protocols	名字戳协议
Plaintext	明文
Encryption function	加密函数
Decryption function	解密函数
Public directory	公共目录
One-way functions	单向函数
Uniform protocol	统一协议
Legitimate user	合法用户
Reduced form	约简形式
Balancing property	平衡性质
Balanced cascade protocol	平衡级联协议
Doubly verified protocol	双重验证协议
Head/Tail	头部/尾部

英文术语	中文翻译
Appending	附加
Name-matching	名字匹配
Deletion	删除
Irreducible string	不可约字符串
Extended word problem	扩展字问题
Transformation rule	转换规则
Cancellation rule	抵消规则
Key safeguarding scheme / Threshold scheme	密钥保护方案 / 阈值方案
Shadows	影子
Polynomial interpolation method	多项式插值法
Lagrange interpolating polynomial	拉格朗日插值多项式
Modular method	模方法
Congruence classes	同余类
Error detection	错误检测
Impatient saboteur	不耐烦的破坏者
Strongly balanced	强平衡的
Linkage property	链接性质