This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

Rethinking about Type-flaw Attacks

Zhiwei Li
Department of SIS
UNC Charlotte
Charlotte, NC 28223
Email: zli19@uncc.edu

Weichao Wang
Department of SIS
UNC Charlotte
Charlotte, NC 28223
Email: weichaowang@uncc.edu

***Abstract*—Many security protocols are vulnerable to type flaw attacks, in which a protocol message may be forged from another message. The previous approaches focus on heuristic schemes to protect specific protocols but fail to expose the enabling factors of such attacks. In this paper, we investigate the relationship between the type flaw attacks on the security protocols and the knowledge of the principals. We formalize the notion of recognizability that characterizes the fact that a message could not be type-flawed. The approach helps us better understand security protocols and gives insights into the detection and prevention of type-flaw attacks.**

I. Introduction

Many security protocols are vulnerable to type flaw attacks, in which a protocol message may be subsequently forged from another message. Let us consider the Otway-Rees protocol [1]:

$$A \rightarrow B : M, A, B, \{N_A, M, A, B\}_{K_{AS}}$$
$$B \rightarrow S : M, A, B, \{N_A, M, A, B\}_{K_{AS}}, \{N_B, M, A, B\}_{K_{BS}}$$
$$S \rightarrow B : M, \{N_A, K_{AB}\}_{K_{AS}}, \{N_B, K_{AB}\}_{K_{BS}}$$
$$B \rightarrow A : M, \{N_A, K_{AB}\}_{K_{AS}}$$

After executing the first three messages, principal A is expecting a K_{AB}, which is a symmetric key shared between A and B, from the trusted third party S. The shared key K_{AB} is dynamically generated by S and A does not have any prior knowledge about the bit string. Therefore, any message of the form $M, \{N_A, t\}_{K_{AS}}$ would be accepted by A, as long as the bit string length of t equals to that of K_{AB}. Thus, an intruder can easily replay the message $\{N_A, M, A, B\}_{K_{AS}}$ to A and then A would use M, A, B as the secret if the length satisfies the requirement.

Various approaches have been proposed to defend against type flaw attacks. Heather et al. [2] propose a tagging scheme to prevent type-flaw attacks, in which tags are used to label each field of a message with its intended type. However, since tag information can potentially be confused with data [3], a tagged protocol may give rise to more intricate attacks. More importantly, the question of whether an existing protocol (without any change) is vulnerable to type-flaw attack is not answered.

Catherine Meadows [4] develops a formal model of types to characterize one's capability to verify messages. Without exploring the intuitive idea behind, the procedure of verifying the locality of types could be rather complicated. In [5], [6], Z specification language is employed to model ambiguous messages. The approach based on Z specification language cannot be directly applied to existing protocol analysis tools in a straight-forward way.

However, most of existing approaches are heuristic without giving a satisfiable answer to the very first question:

Why can a security protocol be type flawed?

Rather than developing one particular defense mechanism against type flaw attacks, in this paper, we pursue to answer this question by exploring a principal's ability/inability to cope with ambiguous messages.

In fact, a protocol could be type-flawed if a message could not be "verified" by the receiver. The question then becomes why the potentially ambiguous message could not be "verified" by the principal.

An intuitive answer to this question might be "because she/he does not know that message". At the first glance, this claim seems to be reasonable. For example, if $Alice$ knows $\{N_B\}_{K_B}$ and an attacker sends out a different message and claims it to be $\{N_B\}_{K_B}$, this attack will not succeed. On the contrary, if $Alice$ does not know $\{N_C\}_{K_B}$, the attacker can fool $Alice$ to accept some other bit string.

However, a closer look at the example shows that even when the receiver does not explicitly know a message, she/he can still verify it. For example, we assume that $Alice$ knows $\{N_B\}_{K_B}$ and Bob's public key K_B. Then even $Alice$ does not know the message N_B, she can still verify whether or not a given message is in fact N_B by simply encrypting it with the public key of Bob (K_B) and compare the result with $\{N_B\}_{K_B}$ which is explicitly in her knowledge.

Thus, verifying a message could be fundamentally different from knowing the message and answering the question is not trivial. Informally, we say a principal is able to *recognize* message t, if she/he has certain expectation about its bit string representation [7]. That is, given a bit string, though she/he may not necessarily know t, she/he can verify whether or not it is the bit she/he can verify whether or not it is the bistring representing the intended message.

To solve this problem, in this paper we first establish the models of the knowledge of the principals and the intruders' capabilities. We define the concept of recognizability based on the equivalence of substitutions. We then use some examples to demonstrate the concept and how it can identify type flaw attacks.

We argue that the investigation of why a security protocol could be type flawed shall gain us a thorough understanding

978-1-4244-5637-6/10/$26.00 ©2010 IEEE

of the vulnerabilities of security protocols and shed important lights on security protocol analysis. As the primary contribution of the paper, we formalize the principal's ability to verify a message by the notion of recognizability. To the best of our knowledge, this is the first formal definition for the study, in the general setting, of principal's ability to verify messages.

The remainder of this paper is organized as follows. In Section II we introduce some background. Section III establishes the models of knowledge and attackers' capabilities. Section IV formally defines recognizability and Section V describes its applications. Section VI discusses related work. Finally, Section VII concludes the paper.

II. Preliminaries

In this section we briefly review the basic definitions of term rewriting systems needed in the rest of the paper. We mainly follow the notations in [8].

A. Term Algebra

We use $fv(t)$ and $fv(T)$ to denote the sets of variables that occur in term t and term set T, respectively. A term is *ground* if $fv(t) = \emptyset$. We discriminate two types of function symbols, namely, *public* and *private* function symbols, denoted by $\mathcal{F}^+$ and $\mathcal{F}^-$, respectively. Public functions are used to describe operations that can be freely performed by a principal, such as encryption and decryption. We point out that decryption operation is conducted even without the proper decryption key and can be applied to any message. It can be different from the decryption of ciphertext. Private functions are used to constrain the relation between terms. For example, public key and private key are described by a private function $\mathtt{kp}$ in this paper.

We say that s is a *subterm* of t, written $s \subseteq t$, if either $s =_s t$ or $t =_s f(t_1, \cdots, t_n)$ and s is a subterm of t_i for some i. We also write $s \subset t$ if $s \subseteq t$ and $s \neq_s t$. We say that a term s *occurs* in a term set T if $s \subseteq u$ for some $u \in T$. For convenience, we will use $ff(t)$ and $sub(t)$ to denote the outmost function symbol of t and the immediate subterm set of a term t [1]. A *context* C is a term with exactly a "hole" $\Box$. Then the term $C[t]$ is C except $\Box$ is replaced by t.

A *substitution* is a finite tuple $[t_1/x_1, ..., t_n/x_n]$ mapping from variables x_i to terms t_i. The *domain* and *range* of a substitution σ is defined by $Dom(\sigma) \stackrel{def}{=} \{x | x\sigma \neq_s x\}$ and $Ran(\sigma) \stackrel{def}{=} \bigcup_{x \in Dom(\sigma)} \{x\sigma\}$, respectively. A substitution σ is *ground* if $fv(Ran(\sigma)) = \emptyset$.

B. Term Rewriting Systems

We write $t_1 =_E t_2$ when equation $t_1 = t_2$ is a logical consequence of equational theory E. To avoid confusion, syntactic equality of two terms t_1 and t_2 will be denoted by $t_1 =_s t_2$. As is commonplace, the reflexive transitive closure of a binary relation $\rightarrow$ is denoted by $\rightarrow^*$.

A *term rewriting system* R consists of a set of rules, $l \rightarrow r$. A term rewriting system R defines a *term rewriting relation* $\rightarrow_R$ in a standard way: $C[l\sigma] \rightarrow_R C[r\sigma]$ where C is a context, $l \rightarrow r \in R$, and σ is a substitution such that $Dom(\sigma) \subseteq fv(l)$[2]. We say that a term s is *reducible* for $\rightarrow_R$ if there is a term t such that $s \rightarrow_R t$ and *irreducible* otherwise. We write $s \rightarrow_R^! t$ if $s \rightarrow_R^* t$ and t is irreducible. If $s \rightarrow_R^! t$, then t is called an *R-normal form* of s. $\rightarrow_R$ is *terminating* if there exists no infinite derivation $t_0 \rightarrow_R t_1 \rightarrow_R \cdots$ and $\rightarrow_R$ is *confluent* if there is a term t such that $t_1 \rightarrow_R^* t$ and $t_2 \rightarrow_R^* t$ whenever $t_0 \rightarrow_R^* t_1$ and $t_0 \rightarrow_R^* t_2$. A term rewriting system R is *convergent* if $\rightarrow_R$ is terminating and confluent. Given an equational theory E, we define term rewriting system R_E by $R_E \stackrel{def}{=} \{l \rightarrow r | l = r \in E\}$ and when R_E is convergent, $t_1 =_E t_2$ iff if t_1 and t_2 have the same R_E-normal form [9], [8].

A substitution σ is *R_E-normal* if all terms in $Ran(\sigma)$ are R_E-normal. We write $\sigma_1 =_E \sigma_2$ to mean $Dom(\sigma_1) = Dom(\sigma_2)$ and $x\sigma_1 =_E x\sigma_2$ for all $x \in Dom(\sigma_1)$. In the rest of this paper, all substitutions are considered R_E-normal.

III. Modeling Knowledge

As noted in the introduction, a principal's ability/inablility to verify messages depends strongly on the nature of the principal's knowledge, though fundamentally different. In this section, we provide a way of defining a principal's knowledge in terms of deducibility. Then, we present the equational theory E_{dy}, used throughout this paper, to model the standard Dolev-Yao intruder.

A. Modeling Knowledge

As we have seen, the idea behind tagging scheme [2] is to distribute more meta information about the message to the recipient, which can be regarded as the auxiliary knowledge specifically constructed for the recipient to recognize messages. Indeed, given more information, a message becomes harder to be forged. It is far from clear what information is suitable for the principal to recognize a message. This explains why message tagging is unnecessary sometimes [10], whereas for some other times it is not sufficient [3].

A principal is able to verify information only if it can be compared, in a mechanized way, with his or her explicit or implicit knowledge. The most straightforward way is to model knowledge in terms of message deducibility [11], [12]. That is, given an equational system E and some messages T one might be able to compute another message t from T under equational theory E. Formally,

$\boxed{\vdash^{(n)}}$

(R1) $\dfrac{t \in T}{T \vdash^{(1)} t}$

(R2) $\dfrac{T \vdash^{(n_1)} t_1 \cdots T \vdash^{(n_k)} t_k}{T \vdash^{(1+\max\limits_{1 \leq i \leq k} n_i)} f(t_1, \cdots, t_k)}\ f \in \mathcal{F}^+$

$\boxed{\vdash_E^{(n)}}$

(R3) $\dfrac{T \vdash^{(n)} t}{T \vdash_E^{(n)} t}$

[1] We let $sub(t) = \{t\}$ and $ff(t) = \emptyset$ if $\|t\| = 1$.

[2] We require $fv(l) \cap fv(C) = \emptyset$; otherwise we could use variable renaming to resolve this conflict

978-1-4244-5637-6/10/$26.00 ©2010 IEEE

$$\text{(R4)} \qquad \frac{T \vdash^{(n)} s \quad s =_E t}{T \vdash_E^{(n+1)} t} s \neq_s t$$

We say that t can be deduced from T, written $T \vdash t$, if $T \vdash^{(n)} t$ for some n. Likewise, t is deduced from T under E, notation $T \vdash_E t$, if $T \vdash_E^{(n)} t$ for some n. As we can see, both $\vdash$ and $\vdash_E$ are closed under substitution.

Lemma III.1. *$T\mu \vdash t$ iff $T \vdash t'$ for some t' such that $t'\mu =_s t$.*

Proof: The "if" part of the lemma is obvious, because $\vdash$ is closed under substitution. We now prove the "only if" part. Suppose that $T\mu \vdash^{(n)} t$. We proceed by induction on n. For the base case, $n = 1$, since t does not contain any common function symbol, by the definition of $\vdash$ we thus have $t \in T\mu$. Then, there is a term $t' \in T$ such that $t'\mu =_s t$. The claim is true. Now, we suppose that $T\mu \vdash^{(n)} t$ implies $T \vdash t'$ for some t' such that $t'\mu =_s t$ whenever $n \leq k$.

For $n = k + 1$, using the definition of $\vdash$ we observe that $T\mu \vdash sub(t)$ and $ff(t) \in \mathcal{F}^+$. Let $t =_s f(t_1, \cdots, t_m)$ and $T\mu \vdash^{(n_i)} t_i$. Since $n_i \leq k$, by induction hypothesis, we know that for each $t_i \in sub(t)$ there exists a term t_i' such that $T \vdash t_i'$ and $t_i'\mu =_s t_i$. Hence, $T \vdash f(t_1', \cdots, t_m')$ and $f(t_1', \cdots, t_m')\mu =_s t$. This completes the proof. ■

B. Modeling Standard Intruders by Equational Theory E_{dy}

In general, $\vdash_E$ may be undecidable without any assumptions on the underlying equational theories. Moreover, it has been shown by Abadi and Cortier [13] that even when equality is decidable $\vdash_E$ could be undecidable. In this paper, we only consider equational theories under which $\vdash_E$ is decidable.

We use the equational theory E_{dy} in Figure 1 to model the standard Dolev-Yao intruder model.

Public signature	$\mathtt{cat}, \mathtt{enc}$
	$\mathtt{fst}, \mathtt{snd}, \mathtt{dec}$
Private signature	$\mathtt{kp}$
Equations E_{dy}	$\mathtt{fst}(\mathtt{cat}(x, y)) = x$
	$\mathtt{snd}(\mathtt{cat}(x, y)) = y$
	$\mathtt{dec}(\mathtt{enc}(x, y), \mathtt{kp}(y)) = x$
	$\mathtt{dec}(\mathtt{enc}(x, \mathtt{kp}(y)), y) = x$
Rewrite Rules R_{dy}	$\mathtt{fst}(\mathtt{cat}(x, y)) \rightarrow x$
	$\mathtt{snd}(\mathtt{cat}(x, y)) \rightarrow y$
	$\mathtt{dec}(\mathtt{enc}(x, y), \mathtt{kp}(y)) \rightarrow x$
	$\mathtt{dec}(\mathtt{enc}(x, \mathtt{kp}(y)), y) \rightarrow x$

Fig. 1. Equational Theory E_{dy} modeling the standard Dolev-Yao intruder.

The equational theory E_{dy} contains two public constructive function symbols (encryption and concatenation), two destructive function symbols (decryption and split), and one private function symbol (key pair). Our analysis does not rely on the actual cryptosystem being used. Rather, we would use $\mathtt{kp}(k)$ to denote the pair key of an encryption key k. To reduce notational clutter, we will often use $\{s\}_t$, $s \cdot t$, and k^- as shorthands for $\mathtt{enc}(s, t)$, $\mathtt{cat}(s, t)$, and $\mathtt{kp}(k)$, respectively.

IV. Defining Recognizability

The purpose of this section is to provide a formal treatment of verifying messages by introducing the notion of recognizability. Before proceeding any further with our general discussion, let us consider again the simple example presented in the introduction.

The initial knowledge of Alice is represented by a ground term set $T_0 = \{K_B, \{N_B\}_{K_B}\}$. She wants to verify whether or not a message is N_B. As explained earlier, this can be done by simply encrypting it with the public key of K_B and compare the result with $\{N_B\}_{K_B}$ which is in her knowledge (i.e., $T_0 \vdash_{E_{dy}} \{N_B\}_{K_B}$).

To illustrate the process of verifying N_B in a more general way, let us bring a free variable x to represent the unverified message and a ground substitution $\sigma_0 = [N_B/x]$ to describe the expected content of x. Meanwhile, we use another substitution σ to indicate possible content value of the unverified x. We often use *expected substitution* and *possible substitution* to refer to σ_0 and σ, respectively.

The process of verifying the content of x becomes the process of exploring all possible σ that satisfies similar properties exhibited by σ_0. For instance, according to Alice the following condition holds:

$$\mathtt{enc}(x\sigma_0, K_B) =_E \{N_B\}_{K_B} \tag{1}$$

Alice can easily detect an unexpected message $x\sigma$ if $\mathtt{enc}(x\sigma, K_B) =_E \{N_B\}_{K_B}$ does not hold true. In other words, the following property

$$\mathtt{enc}(x\sigma, K_B) =_E \{N_B\}_{K_B} \tag{2}$$

is required for all possible forged message $x\sigma$ that can not be distinguished from $x\sigma_0$ by Alice. Finally, it can be easily shown that equation (2) holds only if $x\sigma =_s N_B$.

In summary, it takes two steps for a principal to verify a message:

- Step 1. Identify properties enjoyed by the expected substitution σ_0 and derive equations of the corresponding properties for the possible substitution σ;
- Step 2. Check $\sigma = \sigma_0$ after solving those equations. Intuitively, a message $x\sigma_0$ is recognizable if $\sigma = \sigma_0$ is obtained.

In the next subsection, we will formalize the idea of establishing correspondence between expected substitution and possible substitution (Step 1). Then, we provide a formal definition of recognizability (Step 2).

A. Operational Equivalence

Definition IV.1 (Operational Equivalence)**.** *Let T be a term set and σ_1 and σ_2 be two ground substitutions such that $Dom(\sigma_1) = Dom(\sigma_2) = fv(T)$. They are equivalent in equational theory E w.r.t. term set T, written $\sigma_1 \approx_{E,T} \sigma_2$, if for all terms u and v such that $T \vdash \{u, v\}$ we have $u\sigma_1 =_E v\sigma_1 \Leftrightarrow u\sigma_2 =_E v\sigma_2$.*

The concept of operational equivalence captures the fact that a principal can not distinguish two messages by playing them with messages from the principal's knowledge. To establish recognizability, it is required to ensure that all possible substitutions are identical to the expected substitution.

Example 1. *Consider the term set* $T = \{N_A, K_B^-, x\}$ *and let*

σ_1=$[\{N_A \cdot A\}_{K_B}/x]$ $\quad$ σ_2=$[\{N_A \cdot \{N_B\}_{K_A}\}_{K_B}/x]$
$u =_s \mathtt{fst}(\mathtt{dec}(x, K_B^-))$ $\quad$ $v =_s N_A$

Using the definition of $\vdash$*, it is obvious that* $T \vdash \{u, v\}$*. Moreover,*

$$\begin{aligned} u\sigma_1 &=_s \mathtt{fst}(\mathtt{dec}(\{N_A \cdot A\}_{K_B}, K_B^-)) \\ &\rightarrow_{R_{E_{dy}}} \mathtt{fst}(N_A \cdot A) \\ &=_{E_{dy}} N_A =_s v\sigma_1 \end{aligned}$$

$$\begin{aligned} u\sigma_2 &=_s \mathtt{fst}(\mathtt{dec}(\{N_A \cdot \{N_B\}_{K_A}\}_{K_B}, K_B^-)) \\ &\rightarrow_{R_{E_{dy}}} \mathtt{fst}(N_A \cdot \{N_B\}_{K_A}) \\ &=_{E_{dy}} N_A =_s v\sigma_2 \end{aligned}$$

So, $u\sigma_1 =_{E_{dy}} v\sigma_1$ *and* $u\sigma_2 =_{E_{dy}} v\sigma_2$*. It can be shown that for any* u *and* v *such that* $T \vdash \{u, v\}$ *we have* $u\sigma_1 =_{E_{dy}} v\sigma_1 \Leftrightarrow u\sigma_2 =_{E_{dy}} v\sigma_2$*. That is,* $\sigma_1 \approx_{E_{dy},T} \sigma_2$*.*

This example illustrates how part of a message could be type-flawed. In fact, $\sigma_1 \approx_{E_{dy},T} \sigma$ for any substitution σ satisfying $x\sigma =_s \{N_A \cdot t\}_{K_B}$ and $N_A \not\subseteq t$. This is obvious, because if one explicitly knows N_A (i.e., $T \vdash_{E_{dy}} N_A$), then any message s representing N_A can be easily recognized by comparing bit representations of s and N_A.

The following lemma and theorem give some useful characterizations of operational equivalence.

Lemma IV.2 (Properties)**.** *Let* σ_1 *and* σ_2 *be two ground substitutions.*

(i). $\sigma_1 \approx_{E,T} \sigma_2$ *iff* $\sigma_2 \approx_{E,T} \sigma_1$*;*
(ii). if $\mu\sigma_1 \approx_{E,T} \mu\sigma_2$ *and* $Dom(\sigma_1) = fv(T\mu)$*, then* $\sigma_1 \approx_{E,T\mu} \sigma_2$*;*
(iii). Suppose $T \vdash_E t$*. Then,* $\sigma_1 \approx_{E,T} \sigma_2$ *iff* $\sigma_1 \approx_{E,T\cup\{t\}} \sigma_2$*;*

Proof: (i). Follows immediately from Definition IV.1.

(ii). Without loss of generality, let u and v be two terms such that $T\mu \vdash \{u, v\}$. By Lemma III.1, there exists two terms u' and v' such that $T \vdash \{u', v'\}$, $u'\mu =_s u$, and $v'\mu =_s v$. Moreover, Since $\mu\sigma_1 \approx_{E,T} \mu\sigma_2$ and $T \vdash \{u', v'\}$, we have $u'\mu\sigma_1 =_E v'\mu\sigma_1 \Leftrightarrow u'\mu\sigma_2 =_E v'\mu\sigma_2$. That is, $u\sigma_1 =_E v\sigma_1 \Leftrightarrow u\sigma_2 =_E v\sigma_2$. Moreover, $Dom(\sigma_1) = fv(T\mu)$ by assumption. Using the definition of operational equivalence, we know that $\sigma_1 \approx_{E,T\mu} \sigma_2$.

(iii). The "If" part is trivial. We now prove the "only if" part. To prove $\sigma_1 \approx_{E,T\cup\{t\}} \sigma_2$, it suffices to show that for all terms u and v such that $T \cup \{t\} \vdash \{u, v\}$ we have $u\sigma_1 =_E v\sigma_1 \Leftrightarrow u\sigma_2 =_E v\sigma_2$. Due to the symmetry of σ_1 and σ_2, we only need to prove one direction and proof of the reverse direction can be easily obtained by a similar analysis.

Since $T \vdash_E t$, it is obvious that $T \cup \{t\} \equiv_E T$. Note that $T \vdash \{t\} \vdash \{u, v\}$. By the definition of $\vdash_E$, there exists two terms u' and v' such that $T \vdash \{u', v'\}$, $u' =_E u$, and $v' =_E v$. Clearly, $u\sigma_1 =_E u'\sigma_1$ and $v\sigma_1 =_E v'\sigma_1$. So, $u\sigma_1 =_E v\sigma_1$ implies $u'\sigma_1 =_E v'\sigma_1$. Note that $T \vdash \{u', v'\}$ and $\sigma_1 \approx_{E,T} \sigma_2$. By the definition of operational equivalence, we have $u'\sigma_2 =_E v'\sigma_2$ and thus $u\sigma_2 =_E v\sigma_2$. Likewise, it can be shown that $u\sigma_2 =_E v\sigma_2 \Leftrightarrow u\sigma_1 =_E v\sigma_1$. Hence, $\sigma_1 \approx_{E,T\cup\{t\}} \sigma_2$. ■

B. Recognizability

The last step to establish recognizability is to ensure the possible substitution is identical to the expected substitution.

Definition IV.3 (Recognizability)**.** *Let* T *be a ground term set,* t *be a ground term, and* $\sigma_0 = [t/x]$*. We say that* t *is* recognizable *by term set* T *under equational theory* E *if the following condition holds:*

$$\sigma \approx_{E,T\cup\{x\}} \sigma_0 \text{ iff } \sigma =_E \sigma_0$$

Example 2. *Consider a ground term set*

$$T_0 = \{K_B, \{N_A\}_{K_B}\}$$

Let $\sigma_1 = [K_B^-/x]$*,* $u =_s \mathtt{enc}(\mathtt{dec}(\{N_A\}_{K_B}, x), K_B)$*, and* $v =_s \{N_A\}_{K_B}$*.*

By the definition of $\vdash$*, we have* $T \vdash \{u, v\}$*. Moreover,*

$$\begin{aligned} u\sigma_1 &=_s \mathtt{enc}(\mathtt{dec}(\{N_A\}_{K_B}, K_B^-), K_B) \\ &=_{E_{dy}} \{N_A\}_{K_B} =_s v =_s v\sigma_1 \end{aligned}$$

So, $u\sigma_1 =_{E_{dy}} v\sigma_1$*. Assume that* $\sigma_1' \approx_{E_{dy},T_0\cup\{x\}} \sigma_1$*. Then,* $u\sigma_1' =_{E_{dy}} v\sigma_1'$*. That is,*

$$\mathtt{enc}(\mathtt{dec}(\{N_A\}_{K_B}, x\sigma_1'), K_B) =_{E_{dy}} v\sigma_1' =_s \{N_A\}_{K_B}$$

Now, it is not hard to see that $\mathtt{dec}(\{N_A\}_{K_B}, x\sigma_1') =_{E_{dy}} N_A$ *and thus* $x\sigma_1' =_s K_B^-$*. Note that* $Dom(\sigma_1') = Dom(\sigma_1) = \{x\}$*. Finally, we get* $\sigma_1' = [K_B^-/x] = \sigma_1$*. Similarly, if we let* $\sigma_2 = [N_A/x]$*, it can be shown that* $\sigma_2 \approx_{E_{dy},T_0\cup\{x\}} \sigma_2'$ *iff* $\sigma_2' = \sigma_2$*. Therefore, both* N_A *and* K_B *are recognizable by term set* T_0*.*

In the above example, although neither N_A nor K_B^- is explicitly known (i.e., $T_0 \nvdash_{E_{dy}} \{N_A, K_B^-\}$), one can still recognize them, because for any $\sigma_1' \approx_{E_{dy},T_0\cup\{x\}} \sigma_1$ and $\sigma_2' \approx_{E_{dy},T_0\cup\{x\}} \sigma_2$ we get $\sigma_1' = \sigma_1$ and $\sigma_2' = \sigma_2$, respectively.

V. Application to Labeling Security Protocols

We now apply the concept of recognizability to the analysis of security protocols, or more specifically, to identify messages that are potentially ambiguous.

We start with protocol modeled by strand space model [14] and each incoming message is denoted by a fresh variable. The expected substitution and initial knowledge of protocol participants are set according to the protocol specification. For example, in Otway-Rees protocol, initial knowledge of A is represented by term set $T_a = \{M, A, B, S, N_A, K_{AS}\}$ and the last message received by A is x with an associated expected substitution $\sigma = [\{N_A, K_{AB}\}_{K_{AS}}/x]$.

The analyzed protocol is represented by a set of parameterized strands [15], [16], which is essentially a trace representing protocol execution steps of one particular protocol participant. For example, the final strand of A in Otway-Rees protocol is:

$$\langle +M \cdot A \cdot B \cdot \{N_A \cdot M \cdot A \cdot B\}_{K_{AS}}, -\{N_A \cdot x_1\}_{K_{AS}} \rangle$$

and $\sigma_A = [K_{AB}/x_1]$, in which $+$ and $-$ indicates sending and receiving of a message, respectively. Similarly, the final strand of B in Otway-Rees protocol is:

$$\langle +y_1 \cdot A \cdot B \cdot y_2,\ -y_1 \cdot A \cdot B \cdot y_2 \cdot \{N_B \cdot y_1 \cdot A \cdot B\}_{K_{BS}} + y_1 \cdot y_3 \cdot \{N_B \cdot y_4\}_{K_{BS}},\ -y_1 \cdot y_3 \rangle$$

and $\sigma_B = [M/y_1, \{N_A \cdot M \cdot A \cdot B\}_{K_{AS}}/y_2, \{N_A \cdot K_{AB}\}_{K_{AS}}/y_3, K_{AB}/y_4]$

Now, it becomes apparent from the strand of A that the protocol is vulnerable to type-flaw attack, because $\{N_A \cdot x_1\}_{K_{AS}}$ can be easily forged by the intruder using message $\{N_A \cdot M \cdot A \cdot B\}_{K_{AS}}$.

A similar analysis on the Woo-Lam authentication protocol [17] π_3 shows its vulnerability to type-flaw attacks. The final strand of A in Woo-Lam protocol π_3 is:

$$\langle +A, -x_1, +\{x_1\}_{K_{AS}} \rangle$$

and $\sigma_A = [N_B/x_1]$. The final strand of B is:

$$\langle -A, +N_B, -y_1, +\{A, y_1\}_{K_{BS}}, -\{A, N_B\}_{K_{BS}} \rangle$$

and $\sigma_B = [\{N_B\}_{K_{AS}}/y_1]$.

As claimed in the introduction, we believe reasoning about recognizability helps us understand more thoroughly how a security protocol runs in practice. This is confirmed by our experiment results. Moreover, since our analysis does not change the format of protocol description, analyzed protocol strands can be easily incorporated with security protocol analyzers [18], [19] to detect attacks.

VI. Related Work

The concept of indistinguishability, which comes directly from the classical possible-worlds semantics for knowledge [20], is analog to the possible substitution. Recently, this semantics is generalized for the study of knowledge in cryptographic protocols, which uses static equivalence [21], [22], [23] to capture the indistinguishability for protocol participants. This notion is closely related to our definition of operational equivalence. The most significant difference is that we are concerned about the existence of operation equivalent substitution; while in the study of static equivalence the problem is to determine whether two given substitutions are statically equivalent [13], [24]. Besides, our study is from a cognitive perspective, whereas static equivalence is from a process point of view [21].

VII. Conclusion

In this paper, we provide a theoretical treatment for the study of type-flaw attacks. Our approach differs from previous efforts on detecting type-flaw attacks in that we look into a principal's ability/inability to verify messages. The formal definition of recognizability exposes the enabling factor of the type-flaw attacks and paves the way for the design of future mitigation mechanisms. Immediate extensions to our approach include the design of a decision procedure for recognizability in different attacker models such as Dolev-Yao. The computation cost of the procedure will also be investigated.

Acknowledgment

This research is supported in part by NSF award 0754592.

References

[1] D. Otway and O. Rees, "Efficient and timely mutual authentication," *SIGOPS Oper. Syst. Rev.*, vol. 21, no. 1, pp. 8–10, 1987.

[2] J. Heather, G. Lowe, and S. Schneider, "How to prevent type flaw attacks on security protocols," *J. Comput. Secur.*, vol. 11, no. 2, pp. 217–244, 2003.

[3] C. Meadows, "Identifying potential type confusion in authenticated messages," in *In Proceedings of Foundations of Computer Security 02*, 2002, pp. 75–84.

[4] ——, "A procedure for verifying security against type confusion attacks," in *Computer Security Foundations Workshop, 2003. Proceedings. 16th IEEE*, 30 2003, pp. 62 – 72.

[5] B. W. Long, "Formal verification of type flaw attacks in security protocols," in *APSEC '03: Proceedings of the Tenth Asia-Pacific Software Engineering Conference Software Engineering Conference*. Washington, DC, USA: IEEE Computer Society, 2003, p. 415.

[6] B. Long, C. Fidge, and D. Carrington, "Cross-layer verification of type flaw attacks on security protocols," in *Thirtieth Australasian Computer Science Conference (ACSC2007)*, ser. CRPIT, G. Dobbie, Ed., vol. 62. Ballarat Australia: ACS, 2007, pp. 171–180.

[7] L. Gong, R. Needham, and R. Yahalom, "Reasoning about belief in cryptographic protocols," in *Proc. of IEEE Symposium on Security and Privacy*, 1990, pp. 238–248.

[8] N. Dershowitz and D. A. Plaisted, "Rewriting," in *Handbook of Automated Reasoning*, 2001, pp. 535–610.

[9] G. Birkhoff, "On the structure of abstract algebras," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 31, no. 04, pp. 433–454, 1935.

[10] R. Monroy and J. C. Lopez-Pimentel, "Type flaw attacks."

[11] D. Dolev and A. Yao, "On the security of public key protocols," *Information Theory, IEEE Transactions on*, vol. 29, no. 2, pp. 198–208, Mar 1983.

[12] G. Lowe, "Breaking and fixing the needham-schroeder public-key protocol using fdr," in *TACAs '96: Proceedings of the Second International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, 1996, pp. 147–166.

[13] M. Abadi and V. Cortier, "Deciding knowledge in security protocols under equational theories," *Theor. Comput. Sci.*, vol. 367, no. 1, pp. 2–32, 2006.

[14] F. Fabrega, J. Herzog, and J. Guttman, "Strand spaces: why is a security protocol correct?" in *Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on*, may 1998, pp. 160 –171.

[15] J. Guttman, F. Thayer, J. Carlson, J. Herzog, J. Ramsdell, and B. Sniffen, "Trust management in strand spaces: A rely-guarantee method," in *Proc. of the European Symposium on Programming*, 2004, pp. 325–339.

[16] J. D. Guttman, F. J. Thayer, and L. D. Zuck, "The faithfulness of abstract protocol analysis: message authentication," *J. Comput. Secur.*, vol. 12, no. 6, pp. 865–891, 2004.

[17] T. Y. C. Woo and S. S. Lam, "A lesson on authentication protocol design," *SIGOPS Oper. Syst. Rev.*, vol. 28, no. 3, pp. 24–37, 1994.

[18] D. X. Song, S. Berezin, and A. Perrig, "Athena: a novel approach to efficient automatic security protocol analysis," *J. Comput. Secur.*, vol. 9, no. 1-2, pp. 47–74, 2001.

[19] B. Blanchet, "Automatic verification of correspondences for security protocols," *J. Comput. Secur.*, vol. 17, no. 4, pp. 363–434, 2009.

[20] J. Halpern, "Reasoning about knowledge: a survey," *Handbook of logic in artificial intelligence and logic programming*, vol. 4, pp. 1–34, 1995.

[21] M. Abadi and C. Fournet, "Mobile values, new names, and secure communication," in *Proc. of ACM Symposium on Principles of Programming Languages*, 2001, pp. 104–115.

[22] M. Abadi and P. Rogaway, "Reconciling two views of cryptography (the computational soundness of formal encryption)," *J. Cryptol.*, vol. 20, no. 3, pp. 395–395, 2007.

[23] M. Baudet, V. Cortier, and S. Kremer, "Computationally sound implementations of equational theories against passive adversaries," *Inf. Comput.*, vol. 207, no. 4, pp. 496–520, 2009.

[24] S. Ciobâcă, S. Delaune, and S. Kremer, "Computing knowledge in security protocols under convergent equational theories," in *CADE-22: Proceedings of the 22nd International Conference on Automated Deduction*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 355–370.