Composition of Password-based Protocols

Stéphanie Delaune\textsuperscript{1}, Steve Kremer\textsuperscript{1} and Mark Ryan\textsuperscript{2}

\textsuperscript{1} LSV, ENS de Cachan, CNRS & INRIA, France
\textsuperscript{2} School of Computer Science, University of Birmingham, UK

CSF’08, Pittsburgh
June 2008
Password-based protocols and Guessing attacks

Handshake protocol

A
new r

\[ \text{senc}_w(r) \rightarrow \text{senc}_w(f(r)) \]

B

encrypted key exchange

A
new k

\[ \text{senc}_w(pk(k)) \rightarrow \text{senc}_w(aenc_{pk(k)}(r)) \]

B
new r

Guessing attack on \( w \):

- Guess \( w \)
- Let \( x = \text{sdec}_w(\text{senc}_w(r)) \)
- Let \( y = \text{sdec}_w(\text{senc}_w(f(r))) \)
- Confirm guess of \( w \) by checking \( y = f(x) \)

No guessing attack on \( w \) (assuming it is possible to encode \( pk(k) \) so it looks indistinguishable from a random bitstring).
Password-based protocols and Guessing attacks

Handshake protocol

A

new r

B

\[ \text{senc}_w(r) \]

\[ \text{senc}_w(f(r)) \]

Encrypted key exchange

A

new k

B

\[ \text{senc}_w(pk(k)) \]

\[ \text{senc}_w(aenc_{pk(k)}(r)) \]

Guessing attack on w:

- Guess \( w \)
- Let \( x = \text{sdec}_w(\text{senc}_w(r)) \)
- Let \( y = \text{sdec}_w(\text{senc}_w(f(r))) \)
- Confirm guess of \( w \) by checking \( y = f(x) \)

No guessing attack on \( w \) (assuming it is possible to encode \( pk(k) \) so it looks indistinguishable from a random bitstring).
Password-based protocols and Guessing attacks

Handshake protocol

\[ A \quad \text{new } r \quad B \]

\[ \text{senc}_w(r) \quad \text{senc}_w(f(r)) \]

Guessing attack on \( w \):
- Guess \( w \)
- Let \( x = \text{sdec}_w(\text{senc}_w(r)) \)
- Let \( y = \text{sdec}_w(\text{senc}_w(f(r))) \)
- Confirm guess of \( w \) by checking \( y = f(x) \)

No guessing attack on \( w \) (assuming it is possible to encode \( \text{pk}(k) \) so it looks indistinguishable from a random bitstring).

Encrypted key exchange

\[ A \quad \text{new } k \quad B \]

\[ \text{senc}_w(pk(k)) \quad \text{senc}_w(aenc_{pk(k)}(r)) \]

\[ \text{new } r \]
Composing protocols

Each of them resists guessing attack separately

Attack (even without guessing!) if they are run together:
let \( x = \text{senc}_r(w) \)
Each of them resists guessing attack separately

Attack (even without guessing!) if they are run together:
let $x = \text{senc}_r(w)$
Define guessing attacks in the formal model
  - active and passive attacks

Study composition of protocols that share the password
  - if the individual protocols resist guessing attacks, does the composed protocol also resist?
Describe processes in a simple language inspired by applied pi calculus. Messages are modeled using terms.

- **Abstract algebra** given by a *signature*, i.e. a set of function symbols with arities
- **Equivalence relation** \((=_{E})\) on terms induced by an *equational theory*

### Example (equational theory)

Consider the signature

\[ \Sigma_{\text{enc}} = \{ \text{sdec, senc, adec, aenc, pk, } \langle \rangle, \text{proj}_1, \text{proj}_2 \} \]

<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{sdec}_y(\text{senc}_y(x)) = x )</td>
<td>( \text{sdec}_y(\text{senc}_y(x)) ) equals ( x )</td>
</tr>
<tr>
<td>( \text{senc}_y(\text{sdec}_y(x)) = x )</td>
<td>( \text{senc}_y(\text{sdec}_y(x)) ) equals ( x )</td>
</tr>
<tr>
<td>( \text{adec}<em>y(\text{aenc}</em>{\text{pk}(y)}(x)) = x )</td>
<td>( \text{adec}<em>y(\text{aenc}</em>{\text{pk}(y)}(x)) ) equals ( x )</td>
</tr>
<tr>
<td>( \text{proj}_i(\langle x_1, x_2 \rangle) = x_i )</td>
<td>( \text{proj}_i(\langle x_1, x_2 \rangle) ) equals ( x_i ) for ( i = 1, 2 )</td>
</tr>
</tbody>
</table>
Frames and deduction

As a process evolves, it may output terms which are available to the attacker. The output of a process is called a frame: a set of secrets + a substitution:

\[ \nu \tilde{n}.(\{M_1/x_1\} \mid \{M_2/x_2\} \mid \ldots \mid \{M_n/x_n\}) \]

Example: \( \phi = \nu k, s_1.\{\text{sen}_k(\langle s_1, s_2 \rangle)/x_1, k/x_2\} \)

Definition (Deduction)

\( \nu \tilde{n}.\sigma \vdash_E M \) iff there exists \( N \) such that \( fn(N) \cap \tilde{n} = \emptyset \) and \( N \sigma =_E M \). We call \( N \) a recipe of the term \( M \).

<table>
<thead>
<tr>
<th>Recipe</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi \vdash_{E_{\text{enc}}} k )</td>
</tr>
<tr>
<td>( \phi \vdash_{E_{\text{enc}}} s_1 )</td>
</tr>
<tr>
<td>( \phi \vdash_{E_{\text{enc}}} s_2 )</td>
</tr>
</tbody>
</table>
Frames and deduction

As a process evolves, it may output terms which are available to the attacker. The output of a process is called a frame: a set of secrets + a substitution:

\[ \nu \tilde{n} . (\{ M_1 / x_1 \} \mid \{ M_2 / x_2 \} \mid \ldots \mid \{ M_n / x_n \}) \]

Example: \( \phi = \nu k, s_1 . \{^\text{senc}_k(\langle s_1, s_2 \rangle) / x_1, k / x_2 \} \)

**Definition (Deduction)**

\( \nu \tilde{n} . \sigma \vdash_E M \) iff there exists \( N \) such that \( fn(N) \cap \tilde{n} = \emptyset \) and \( N \sigma =_E M \). We call \( N \) a *recipe* of the term \( M \).

<table>
<thead>
<tr>
<th>Recipe</th>
</tr>
</thead>
</table>
| \( \phi \vdash_{E_{\text{enc}}} k \) | \( x_2 \) 
| \( \phi \vdash_{E_{\text{enc}}} s_1 \) | \( \text{proj}_1(\text{sdec}_{x_2}(x_1)) \) 
| \( \phi \vdash_{E_{\text{enc}}} s_2 \) | \( s_2 \) |
Definition (Static equivalence)

Two frames are statically equivalent if there is no “test” that tells them apart.

\( \phi \) and \( \psi \) are statically equivalent, \( \phi \approx_{E} \psi \), when:
- \( \text{dom}(\phi_1) = \text{dom}(\phi_2) \), and
- for all terms \( M, N \) such that \( \tilde{n} \cap (\text{fn}(M) \cup \text{fn}(N)) = \emptyset \),
  \[ M\phi =_{E} N\phi \iff M\psi =_{E} N\psi \]

Example

\[ \phi = \nu k.\{\text{senc}_k(s_0)/x_1, k/x_2\} \not\approx \nu k.\{\text{senc}_k(s_1)/x_1, k/x_2\} = \phi' \]

because of the test \((\text{sdec}_{x_2}(x_1), s_0)\)

However,

\[ \nu k.\{\text{senc}_k(s_0)/x_1\} \approx \nu k.\{\text{senc}_k(s_1)/x_1\} \]
Guessing attacks (passive case)

A passive guessing or dictionary attack consists of two phases:

1. The attacker eavesdrops on one or several sessions of a protocol.
2. The attacker tries offline each of the possible passwords (e.g. using a dictionary) on the data collected during the first phase.

We suppose the eavesdropping phase results in a frame $\nu w.\phi$.

**Definition (Passive guessing attacks)**

$\nu w.\phi$ is resistant to guessing attacks against $w$ iff

$$\nu w.(\phi \mid \{^w/x\}) \approx \nu w.(\phi \mid \nu w'.\{^{w'}/x\})$$

[Baudet05, Corin et al.03]
EKE resists guessing attacks?

EKE resists guessing attacks only if $pk(k)$ can be encoded indistinguishably from an arb. bitstring.

Consider the equational theory:

\[
\begin{align*}
sdec_y(senc_y(x)) &= x \\
senc_y(sdec_y(x)) &= x \\
\text{adec}_y(aenc_{pk(y)}(x)) &= x \\
\text{proj}_i(\langle x_1, x_2 \rangle) &= x_i \ (i = 1, 2)
\end{align*}
\]

We have

\[
\nu w, k.\left(\left\{senc_w(pk(k)) / x_1 \right\}, \left\{w / x_2 \right\}\right) \approx \nu w, w', k.\left(\left\{senc_w(pk(k)) / x_1 \right\}, \left\{w' / x_2 \right\}\right)
\]
EKE resists guessing attacks?

EKE resists guessing attacks only if $pk(k)$ can be encoded indistinguishably from an arb. bitstring.

Consider the equational theory:

\[
\begin{align*}
\text{sdec}_y(\text{senc}_y(x)) &= x \\
\text{senc}_y(\text{sdec}_y(x)) &= x \\
\text{adec}_y(\text{aenc}_{pk(y)}(x)) &= x \\
\text{proj}_i(\langle x_1, x_2 \rangle) &= x_i \ (i = 1, 2) \\
\text{ispk}(pk(x)) &= \text{true}
\end{align*}
\]

We have

\[
\nu w, k. (\{\text{senc}_w(pk(k))/x_1\}, \{w/x_2\}) \not\approx \nu w, w', k. (\{\text{senc}_w(pk(k))/x_1\}, \{w'/x_2\})
\]

as witnessed by the test: $\text{ispk}(\text{sdec}_{x_2}(x_1)) = \text{true}$. 
Composing protocols that are resistant to passive guessing attacks

**Proposition**

The three following statements are equivalent:

1. $\nu w.\phi \ | \ \{^w/x\} \approx \nu w.\phi \ | \ \nu w'.\{^w'/x\}$  
   [Baudet05]
2. $\phi \approx \nu w.\phi$  
   [Corin et al.03]
3. $\phi \approx \phi\{^w'/w\}$

**Corollary**

If $\nu w.\phi_1$ and $\nu w.\phi_2$ are resistant to guessing attacks against $w$ then $\nu w.(\phi_1 \ | \ \phi_2)$ is also resistant to guessing attacks against $w$.

Thus, resistance to guessing attacks composes in the passive case. In particular, resistance for one session implies resistance for multiple sessions.
Composing protocols that are resistant to passive guessing attacks

Proposition

The three following statements are equivalent:

1. \( \nu w. \phi \mid \{^w/x\} \approx \nu w. \phi \mid \nu w'.\{^w'/x\} \)  
   [Baudet05]

2. \( \phi \approx \nu w. \phi \)  
   [Corin et al.03]

3. \( \phi \approx \phi\{^w'/w\} \)

Corollary

If \( \nu w.\phi_1 \) and \( \nu w.\phi_2 \) are resistant to guessing attacks against \( w \) then \( \nu w.(\phi_1 \mid \phi_2) \) is also resistant to guessing attacks against \( w \).

Thus, resistance to guessing attacks composes in the passive case. In particular, resistance for one session implies resistance for multiple sessions.
Active case
Syntax of the process language

\[ P, Q, R := \]

- 0: null process
- \( P \mid Q \): parallel composition
- \( \text{in}(x).P \): message input
- \( \text{out}(M).P \): message output
- \( \text{if } M = N \text{ then } P \text{ else } Q \): conditional

Extended processes: \( A, B, C := P \mid A \mid B \mid \nu n.A \mid \{ M/x \} \)

Example: “EKE++”

\[ \nu w. (\nu k. (\text{out}(\text{senc}_w(pk(k))).\text{in}(x).\text{out}(\text{senc}_{\text{aenc}_k}(\text{sdec}_w(x)))(w)) \mid \text{in}(y).\nu r.\text{out}(\text{senc}_w(\text{aenc}_y(r))).\text{in}(z) \ldots .) \]
Semantics of the process language

**Structural equivalence:** the smallest equivalence relation closed by application of evaluation contexts and such that

\[
\begin{align*}
\text{Par-0} & \quad A \mid 0 \equiv A \\
\text{Par-C} & \quad A \mid B \equiv B \mid A \\
\text{Par-A} & \quad (A \mid B) \mid C \equiv A \mid (B \mid C) \\
\text{New-Par} & \quad A \mid \nu n.B \equiv \nu n.(A \mid B) \\
\text{New-C} & \quad \nu n_1.\nu n_2.A \equiv \nu n_2.\nu n_1.A
\end{align*}
\]

Operational semantics: smallest relation between extended processes which is closed under structural equivalence (\(\equiv\)) and such that

\[
\begin{align*}
\text{In} & \quad \text{in}(x).P & \overset{\text{in}(M)}{\longrightarrow} & P\{^M/_{x}\} \\
\text{Out} & \quad \text{out}(M).P & \overset{\text{out}(M)}{\longrightarrow} & P \mid \{^M/_{x}\} & \text{where } x \text{ is a fresh variable} \\
\text{Then} & \quad \text{if } M = N \text{ then } P \text{ else } Q & \overset{\tau}{\longrightarrow} & P \\
\text{Else} & \quad \text{if } M = N \text{ then } P \text{ else } Q & \overset{\tau}{\longrightarrow} & Q & \text{where } M \not=_{E} N \\
\text{Cont.} & \quad A \overset{\ell}{\rightarrow} B & \quad C[A] \overset{\ell}{\rightarrow} C[B] & \text{where } C \text{ is an evaluation context} \\
& \quad & & \text{if } \ell = \text{in}(M) \text{ then } \phi(C[A]) \vdash_{E} M
\end{align*}
\]
Semantics of the process language

**Structural equivalence:** the smallest equivalence relation closed by application of evaluation contexts and such that

- **Par-0** \( A \parallel 0 \equiv A \)
- **New-Par** \( A \parallel \nu n . B \equiv \nu n . (A \parallel B) \)
- **Par-C** \( A \parallel B \equiv B \parallel A \)
- **Par-A** \( (A \parallel B) \parallel C \equiv A \parallel (B \parallel C) \)
- **New-C** \( \nu n_1 . \nu n_2 . A \equiv \nu n_2 . \nu n_1 . A \)

**Operational semantics:** smallest relation between extended processes which is closed under structural equivalence (\( \equiv \)) and such that

\[
\begin{align*}
\text{In} & \quad \text{in}(x).P \xrightarrow{\text{in}(M)} P\{M/x\} \\
\text{Out} & \quad \text{out}(M).P \xrightarrow{\text{out}(M)} P \mid \{M/x\} \quad \text{where } x \text{ is a fresh variable} \\
\text{Then} & \quad \text{if } M = N \text{ then } P \text{ else } Q \xrightarrow{\tau} P \quad \text{where } M \equiv_E N \\
\text{Else} & \quad \text{if } M = N \text{ then } P \text{ else } Q \xrightarrow{\tau} Q \quad \text{where } M \not\equiv_E N \\
\text{Cont.} & \quad A \xrightarrow{\ell} B \\
& \quad \frac{C[A] \xrightarrow{\ell} C[B]}{}
\end{align*}
\]

where \( C \) is an evaluation context

if \( \ell = \text{in}(M) \) then \( \phi(C[A]) \vdash_E M \)
Consider the handshake protocol. In our calculus it is modelled as:

- \( A = \nu n. \text{out}(\text{senc}_w(n)). \text{in}(x). \text{if } \text{sdec}_w(x) = f(n) \text{ then } P \)
- \( B = \text{in}(y). \text{out}(\text{senc}_w(f(\text{sdec}_w(y)))) \)

which admits the execution

\[
\nu w. (A | B) \quad \begin{array}{c}
\nu w. \nu n. (B | \{\text{senc}_w(n)/x_1\} | \text{in}(x). \text{if } \text{sdec}_w(x) = f(n) \text{ then } P)
\\
\nu w. \nu n. (\text{out}(M) | \{\text{senc}_w(n)/x_1\} | \text{in}(x). \text{if } \text{sdec}_w(x) = f(n) \text{ then } P)
\\
\nu w. \nu n. (\{\text{senc}_w(n)/x_1\} | \{M/x_2\} | \text{in}(x). \text{if } \text{sdec}_w(\text{senc}_w(f(n))) = f(n) \text{ then } P)
\\
\tau \quad \nu w. \nu n. (\{\text{senc}_w(n)/x_1\} | \{M/x_2\} | P)
\end{array}
\]

where \( M = \text{senc}_w(f(\text{sdec}_w(\text{senc}_w(n)))) = E \text{senc}_w(f(n)) \)
Guessing attacks (active case)

Definition (Active guessing attacks)

A is resistant to guessing attack against \( w \) if, for every process \( B \) such that \( A \to^* B \), we have that \( \phi(B) \) is resistant to guessing attacks against \( w \).

Frame of a process

\( \phi(A) = \) result of replacing plain processes in \( A \) by 0.
Composing protocols that are resistant to active guessing attacks

Contrary to passive case, resistance does not compose in general.

After the execution in which \( x = \text{senc}_r(w) \):

\[
\phi = \nu w, k, r. (\{\text{senc}_w(pk(k)) / x_1\}, \{\text{senc}_w(aenc_{pk(k)}(r)) / x_2\}, \{\text{senc}_r(w) / x_3\}, \{w / x_4\})
\]
Intuitively, a protocol is well-tagged w.r.t. a secret $w$ if all the occurrences of $w$ are of the form $h(\alpha, w)$

**Definition (well-tagged)**

$M$ is $\alpha$-tagged w.r.t. $w$ if there exists $M'$ s.t. $M'\{^{h(\alpha,w)/w}_w\} =_E M$.

A term is said well-tagged w.r.t. $w$ if it is $\alpha$-tagged for some name $\alpha$.

$A$ is $\alpha$-tagged if any term occurring in it is $\alpha$-tagged. An extended process is well-tagged if it is $\alpha$-tagged for some name $\alpha$.

Well-tagged processes compose!

**Theorem (composition result)**

*Let $A_1$ be $\alpha$-tagged and $A_2$ be $\beta$-tagged w.r.t. $w$.*

*If $\nu w.A_1$ and $\nu w.A_2$ are resistant to guessing attacks against $w$ then $\nu w.(A_1 | A_2)$ is also resistant to guessing attacks against $w$.*
Well-tagged protocols and composition

Intuitively, a protocol is well-tagged w.r.t. a secret $w$ if all the occurrences of $w$ are of the form $h(\alpha, w)$

Definition (well-tagged)

$M$ is $\alpha$-tagged w.r.t. $w$ if there exists $M'$ s.t. $M'\{^h(\alpha, w)/w\} =_E M$.

A term is said well-tagged w.r.t. $w$ if it is $\alpha$-tagged for some name $\alpha$.

$A$ is $\alpha$-tagged if any term occurring in it is $\alpha$-tagged. An extended process is well-tagged if it is $\alpha$-tagged for some name $\alpha$.

Well-tagged processes compose!

Theorem (composition result)

Let $A_1$ be $\alpha$-tagged and $A_2$ be $\beta$-tagged w.r.t. $w$.

If $\nu w.A_1$ and $\nu w.A_2$ are resistant to guessing attacks against $w$ then $\nu w.(A_1 \mid A_2)$ is also resistant to guessing attacks against $w$. 
A secure transformation

**Theorem**

If $\nu w. A$ is resistant to guessing attacks against $w$ then $\nu w. (A\{h(\alpha, w)/w\})$ is also resistant to guessing attacks against $w$.

Easy, syntactic transformation: thumbrule for good design?

**Remark on other transformations:**

- replacing $w$ by $\langle w, \alpha \rangle$ does not guarantee composition
- tagging encryptions (used in [CortierDelaitreDelaune07] to ensure composition of other properties) would add guessing attacks
Conclusion and future work

**Passive** guessing attacks **do compose**.

**Active** guessing attacks **do not compose in general**.

But for **well-tagged protocols**:

- **Secure transformation** to obtain well-tagged protocols

Future work

Avoid tags: are there (interesting) **classes of protocols and equational theories** for which guessing attacks compose?

Other **forms of composition**:

- composition for **observational equivalence**
- **sequential** composition