Composition of Password-based Protocols

Stéphanie Delaune[1], Steve Kremer[1] and Mark Ryan[2]

[1] LSV, ENS de Cachan, CNRS & INRIA, France

[2] School of Computer Science, University of Birmingham, UK

CSF'08, Pittsburgh
June 2008

Password-based protocols and Guessing attacks

Handshake protocol

A		B
new r	$\xrightarrow{\text{senc}_w(r)}$
	$\xleftarrow{\text{senc}_w(f(r))}$

Encrypted key exchange

A		B
new k	$\xrightarrow{\text{senc}_w(pk(k))}$
		new r
	$\xleftarrow{\text{senc}_w(aenc_{pk(k)}(r))}$

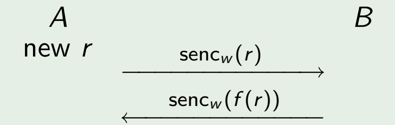
Guessing attack on w:

- 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 $y = f(x)$

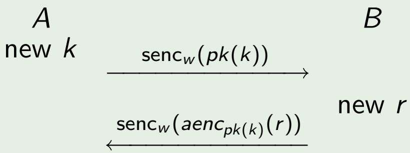
No guessing attack on w (assuming it is possible to encode $pk(k)$ so it looks indistinguishable from a random bitstring).

Password-based protocols and Guessing attacks

Handshake protocol



Encrypted key exchange



Guessing attack on w:

- Guess w
- Let $x = \mathrm{sdec}_w(\mathrm{senc}_w(r))$
- Let $y = \mathrm{sdec}_w(\mathrm{senc}_w(f(r)))$
- 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 B

new r

$A \xrightarrow{\mathsf{senc}_w(r)} B$

$A \xleftarrow{\mathsf{senc}_w(f(r))} B$

Guessing attack on w:

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

Encrypted key exchange

A B

new k

$A \xrightarrow{\mathsf{senc}_w(pk(k))} B$

new r (B)

$A \xleftarrow{\mathsf{senc}_w(aenc_{pk(k)}(r))} B$

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

Composing protocols

"EKE++"

A — new k | B

$A \xrightarrow{\text{senc}_w(pk(k))} B$

B: new r

$B \xrightarrow{\text{senc}_w(aenc_{pk(k)}(r))} A$

$A \xrightarrow{\text{senc}_r(w)} B$

"EKE+++"

A — new k | B

$A \xrightarrow{\text{senc}_w(pk(k))} B$

B: new r

$B \xrightarrow{\text{senc}_w(aenc_{pk(k)}(r))} A$

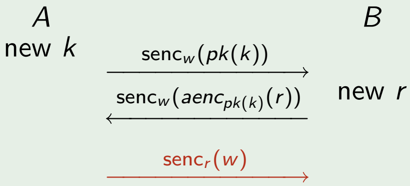
$\xrightarrow{x} B$

$B \xrightarrow{\text{sdec}_r(x)}$

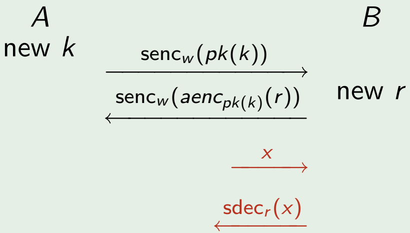
- Each of them resists guessing attack separately
- Attack (even without guessing!) if they are run together: let $x = \text{senc}_r(w)$

Composing protocols

"EKE++"



"EKE+++"



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

Outline

- 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?

Terms and equational theories

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
$\Sigma_{\mathsf{enc}} = \{\mathsf{sdec}, \mathsf{senc}, \mathsf{adec}, \mathsf{aenc}, \mathsf{pk}, \langle\ \rangle, \mathsf{proj}_1, \mathsf{proj}_2\}$

$$\begin{array}{rcl@{\qquad}rcl} \mathsf{sdec}_y(\mathsf{senc}_y(x)) & = & x & \mathsf{adec}_y(\mathsf{aenc}_{\mathsf{pk}(y)}(x) & = & x \\ \mathsf{senc}_y(\mathsf{sdec}_y(x)) & = & x & \mathsf{proj}_i(\langle x_1, x_2\rangle) & = & x_i \quad i = 1, 2 \end{array}$$

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

Definition (Deduction)

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

	Recipe
$\phi \vdash_{\mathsf{E_{enc}}} k$	x_2
$\phi \vdash_{\mathsf{E_{enc}}} s_1$	$\mathsf{proj}_1(\mathsf{sdec}_{x_2}(x_1))$
$\phi \vdash_{\mathsf{E_{enc}}} s_2$	s_2

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

Definition (Deduction)

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

	Recipe
$\phi \vdash_{\mathsf{E}_{\mathsf{enc}}} k$	x_2
$\phi \vdash_{\mathsf{E}_{\mathsf{enc}}} s_1$	$\mathrm{proj}_1(\mathrm{sdec}_{x_2}(x_1))$
$\phi \vdash_{\mathsf{E}_{\mathsf{enc}}} s_2$	s_2

Static equivalence

Definition (Static equivalence)

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

ϕ and ψ are statically equivalent, $\phi \approx_{\mathsf{E}} \psi$, when:

- $dom(\phi_1) = dom(\phi_2)$, and
- for all terms M, N such that $\tilde{n} \cap (fn(M) \cup fn(N)) = \emptyset$, $M\phi =_{\mathsf{E}} N\phi$ iff $M\psi =_{\mathsf{E}} N\psi$

Example

$$\phi = \nu k.\{^{\mathsf{senc}_k(s_0)}/_{x_1}, {}^{k}/_{x_2}\} \not\approx \nu k.\{^{\mathsf{senc}_k(s_1)}/_{x_1}, {}^{k}/_{x_2}\} = \phi'$$

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

However,

$$\nu k.\{^{\mathsf{senc}_k(s_0)}/_{x_1}\} \approx \nu k.\{^{\mathsf{senc}_k(s_1)}/_{x_1}\}$$

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 $\nu w.\phi$.

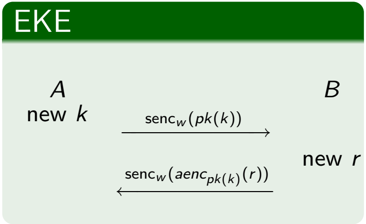
Definition (Passive guessing attacks)

$\nu w.\phi$ 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:

$$
\begin{aligned}
\mathrm{sdec}_y(\mathrm{senc}_y(x)) &= x \\
\mathrm{senc}_y(\mathrm{sdec}_y(x)) &= x \\
\mathrm{adec}_y(\mathrm{aenc}_{\mathrm{pk}(y)}(x)) &= x \\
\mathrm{proj}_i(\langle x_1, x_2 \rangle) &= x_i \ (i = 1, 2)
\end{aligned}
$$

We have

$$\nu w, k.(\{^{\mathrm{senc}_w(pk(k))}/_{x_1}\}, \{^{w}/_{x_2}\}) \approx \nu w, w', k.(\{^{\mathrm{senc}_w(pk(k))}/_{x_1}\}, \{^{w'}/_{x_2}\})$$

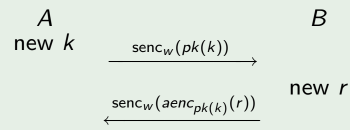
EKE resists guessing attacks?

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

Consider the equational theory:

$$
\begin{array}{rcl}
\mathrm{sdec}_y(\mathrm{senc}_y(x)) & = & x \\
\mathrm{senc}_y(\mathrm{sdec}_y(x)) & = & x \\
\mathrm{adec}_y(\mathrm{aenc}_{\mathrm{pk}(y)}(x) & = & x \\
\mathrm{proj}_i(\langle x_1, x_2 \rangle) & = & x_i \ (i = 1, 2) \\
\mathrm{ispk}(\mathrm{pk}(x)) & = & \mathrm{true}
\end{array}
$$

EKE



We have

$$\nu w, k.(\{^{\mathrm{senc}_w(pk(k))}/_{x_1}\}, \{^{w}/_{x_2}\}) \not\approx \nu w, w', k.(\{^{\mathrm{senc}_w(pk(k))}/_{x_1}\}, \{^{w'}/_{x_2}\})$$

as witnessed by the test: $\mathrm{ispk}(\mathrm{sdec}_{x_2}(x_1)) = \mathrm{true}$.

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 resitance 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 resitance for multiple sessions.

Active case

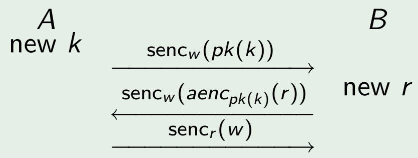
Syntax of the process language

$P, Q, R :=$ Plain processes

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

Extended processes $A, B, C := P \mid A \mid B \mid \nu n.A \mid \{^M/_x\}$

Example: "EKE++"



$$
\begin{aligned}
&\nu w.(\\
&\quad \nu k.(\mathsf{out}(\mathsf{senc}_w(pk(k))).\mathsf{in}(x). \\
&\quad \mathsf{out}(\mathsf{senc}_{\mathsf{adec}_k(\mathsf{sdec}_w(x))}(w)) \\
&\mid \\
&\quad \mathsf{in}(y).\nu r.\mathsf{out}(\mathsf{senc}_w(\mathsf{aenc}_y(r))). \\
&\quad \mathsf{in}(z).\ldots \\
&)
\end{aligned}
$$

Semantics of the process language

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

PAR-0	$A \mid 0$	$\equiv$	A	NEW-PAR	$A \mid \nu n.B$	$\equiv$	$\nu n.(A \mid B)$
PAR-C	$A \mid B$	$\equiv$	$B \mid A$				$n \notin fn(A)$
PAR-A	$(A \mid B) \mid C$	$\equiv$	$A \mid (B \mid C)$	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

IN	$\text{in}(x).P \xrightarrow{in(M)} P\{^M/_x\}$	
OUT	$\text{out}(M).P \xrightarrow{out(M)} P \mid \{^M/_x\}$	where x is a fresh variable
THEN	if $M = N$ then P else $Q \xrightarrow{\tau} P$	where $M =_E N$
ELSE	if $M = N$ then P else $Q \xrightarrow{\tau} Q$	where $M \neq_E N$
CONT.	$\dfrac{A \xrightarrow{\ell} B}{C[A] \xrightarrow{\ell} C[B]}$	where C is an evaluation context if $\ell = in(M)$ then $\phi(C[A]) \vdash_E M$

Semantics of the process language

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

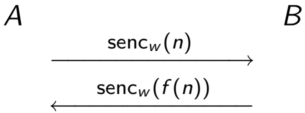
PAR-0	$A \mid 0 \equiv A$		NEW-PAR	$A \mid \nu n.B \equiv \nu n.(A \mid B)$		
PAR-C	$A \mid B \equiv B \mid A$			$n \notin fn(A)$		
PAR-A	$(A \mid B) \mid C \equiv A \mid (B \mid C)$		NEW-C	$\nu n_1.\nu n_2.A \equiv \nu n_2.\nu n_1.A$		

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

IN	$\mathrm{in}(x).P \xrightarrow{in(M)} P\{^M/_x\}$	
OUT	$\mathrm{out}(M).P \xrightarrow{out(M)} P \mid \{^M/_x\}$	where x is a fresh variable
THEN	if $M = N$ then P else $Q \xrightarrow{\tau} P$	where $M =_{\mathsf{E}} N$
ELSE	if $M = N$ then P else $Q \xrightarrow{\tau} Q$	where $M \neq_{\mathsf{E}} N$
CONT.	$$\frac{A \xrightarrow{\ell} B}{C[A] \xrightarrow{\ell} C[B]}$$	where C is an evaluation context if $\ell = in(M)$ then $\phi(C[A]) \vdash_{\mathsf{E}} M$

Example

Consider the handshake protocol. In our calculus it is modelled as:



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

which admits the execution

$$
\begin{array}{ll}
\nu w.(A \mid B) & \\
\xrightarrow{out(\mathsf{senc}_w(n))} & \nu w.\nu n.(B \mid \{^{\mathsf{senc}_w(n)}/_{x_1}\} \mid \mathsf{in}(x).\ \mathsf{if}\ \mathsf{sdec}_w(x) = f(n)\ \mathsf{then}\ P) \\
\xrightarrow{in(\mathsf{senc}_w(n))} & \nu w.\nu n.(\mathsf{out}(M) \mid \{^{\mathsf{senc}_w(n)}/_{x_1}\} \mid \mathsf{in}(x).\ \mathsf{if}\ \mathsf{sdec}_w(x) = f(n)\ \mathsf{then}\ P) \\
\xrightarrow{out(M)} & \nu w.\nu n.(\{^{\mathsf{senc}_w(n)}/_{x_1}\} \mid \{^{M}/_{x_2}\} \mid \mathsf{in}(x).\ \mathsf{if}\ \mathsf{sdec}_w(x) = f(n)\ \mathsf{then}\ P) \\
\xrightarrow{in(\mathsf{senc}_w(f(n)))} & \nu w.\nu n.(\{^{\mathsf{senc}_w(n)}/_{x_1}\} \mid \{^{M}/_{x_2}\} \mid\ \mathsf{if}\ \mathsf{sdec}_w(\mathsf{senc}_w(f(n))) = f(n) \\
 & \qquad \mathsf{then}\ P) \\
\xrightarrow{\tau} & \nu w.\nu n.(\{^{\mathsf{senc}_w(n)}/_{x_1}\} \mid \{^{M}/_{x_2}\} \mid P)
\end{array}
$$

where $M = \mathsf{senc}_w(f(\mathsf{sdec}_w(\mathsf{senc}_w(n)))) =_{\mathsf{E}} \mathsf{senc}_w(f(n))$

Guessing attacks (active case)

Definition (Active guessing attacks)

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

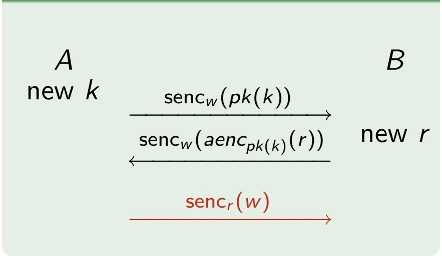
Frame of a process

$\phi(A)$ = result of replacing plain processes in A by 0.

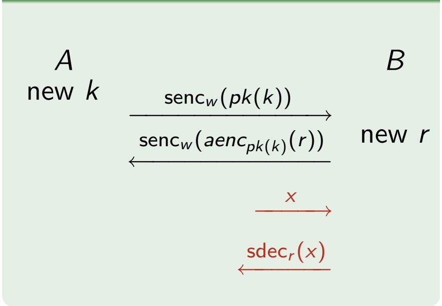
Composing protocols that are resistant to active guessing attacks

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

“EKE++”



“EKE+++”



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

$$\phi = \nu w, k, r.(\quad \{^{\mathrm{senc}_w(pk(k))}/_{x_1}\}, \{^{\mathrm{senc}_w(aenc_{pk(k)}(r))}/_{x_2}\}, \{^{\mathrm{senc}_r(w)}/_{x_3}\}, \{^{w}/_{x_4}\})$$

Well-taged 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 $\mathsf{h}(\alpha, w)$

Definition (well-tagged)

M is α-tagged w.r.t. w if there exists M' s.t. $M'\{^{\mathsf{h}(\alpha,w)}/_{w}\} =_{\mathsf{E}} M$.
A term is said well-tagged w.r.t. w if it is α-tagged for some name α.
A is α-tagged if any term occurring in it is α-tagged. An extended process is well-tagged if it is α-tagged for some name α.

Well-tagged processes compose!

Theorem (composition result)

*Let A_1 be α-tagged and A_2 be β-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.*

Well-taged 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 $\mathsf{h}(\alpha, w)$

Definition (well-tagged)

M is α-tagged w.r.t. w if there exists M' s.t. $M'\{^{\mathsf{h}(\alpha,w)}/_{w}\} =_{\mathsf{E}} M$.
A term is said well-tagged w.r.t. w if it is α-tagged for some name α.
A is α-tagged if any term occurring in it is α-tagged. An extended process is well-tagged if it is α-tagged for some name α.

Well-tagged processes compose!

Theorem (composition result)

Let A_1 be α-tagged and A_2 be β-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 $\nu w.(A\{^{\mathsf{h}(\alpha,w)}/_{w}\})$ 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-taged 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