For issuing certificates we assume an authentication server S with public and private keys K_S and K_S^{-1} respectively.

Certificates are of the form $cert(A, K) = \{A, K\}_{K_S^{-1}}$.

For issuing certificates we assume an authentication server S with public and private keys K_S and K_S^{-1} respectively.

Certificates are of the form $cert(A, K) = \{A, K\}_{K_S^{-1}}$.

$$sessionK : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \times \{0, 1\} \to \mathbb{N}$$
$$clientK(x, y, z) = sessionK(x, y, z, 0)$$
$$serverK(x, y, z) = sessionK(x, y, z, 1)$$

Functions $sessionK$ and PRF are assumed to be collision-free.

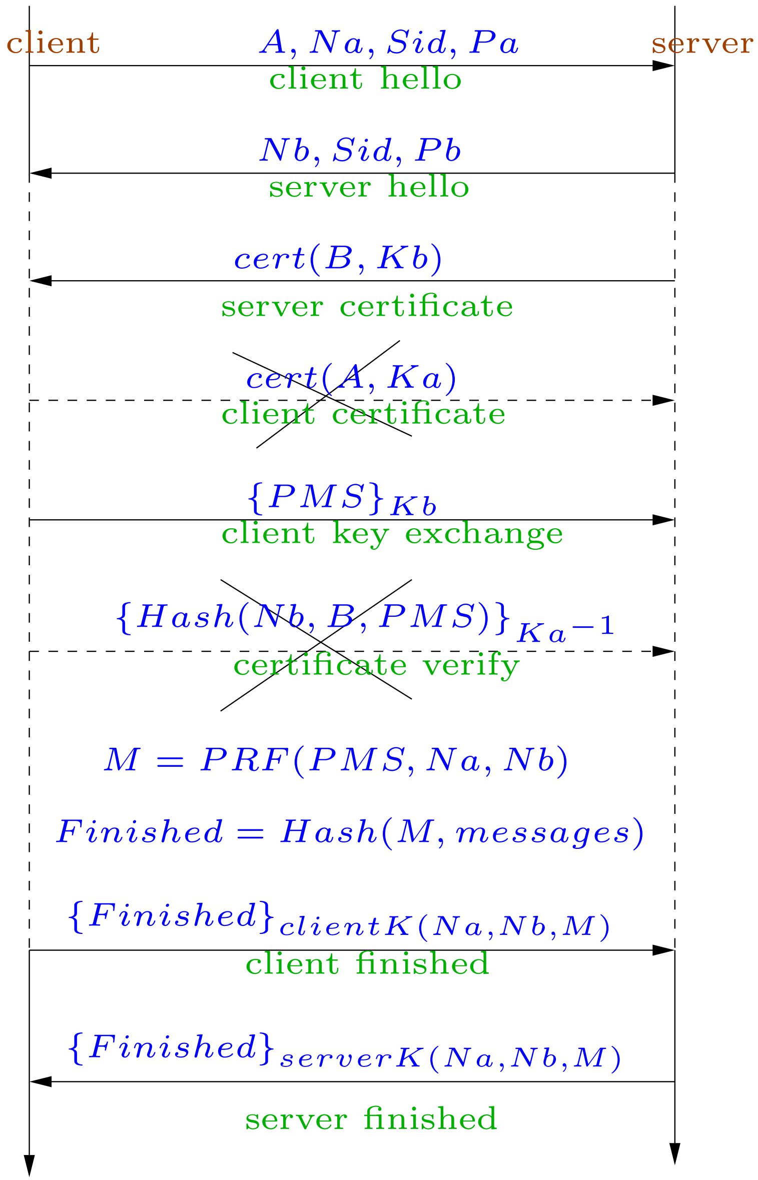
Session resumption

A session is resumed by using the corresponding session identifier Sid and master secret M.

Fresh nonces Na and Nb need to be exchanged.

The messages exchanged for session resumption are client hello, server hello, client finished and server finished.

Session resumption is supposed to be secure even if old keys from the same session are compromised.



The client certificate and certificate verify messages are optional.

Hence A can remain unauthenticated, leading to an attack where the intruder pretends to be another client.

Ensuring correctness of a cryptographic protocol

- Finding some attacks (with a tool)
- Guaranteeing that there are no attacks (certification)
  - Writing proofs checked by a tool like Isabelle: reliable, but costs time and human effort.
  - Use an automated tool to guarantee correctness: fast but works only for specific classes of protocols.

Ensuring correctness of a cryptographic protocol

- Finding some attacks (with a tool)
- Guaranteeing that there are no attacks (certification)
  - Writing proofs checked by a tool like Isabelle: reliable, but costs time and human effort.
  - Use an automated tool to guarantee correctness: fast but works only for specific classes of protocols.

Paulson's proof of the TLS handshake protocol in Isabelle:
6 weeks of human time, a dozen pages long proof script,
proof checking by Isabelle in a few minutes.

Available at: `http://www.cl.cam.ac.uk/Research/HVG/Isabelle/dist/library/HOL/Auth/TLS.html`

Automatically analyzing cryptographic protocols

- Consider a fixed number of sessions (to detect some attacks)
  - Passive intruder: checking secrecy mounts to solving the intruder deduction problem.
  - Active intruder
- Infinitely many sessions (to certify protocols):

Automatically analyzing cryptographic protocols

- Consider a fixed number of sessions (to detect some attacks)
  - Passive intruder: checking secrecy mounts to solving the intruder deduction problem.
  - Active intruder
- Infinitely many sessions (to certify protocols):
  Need to consider restricted classes of protocols

Ping-Pong Protocols

A simple class of protocols whose security can be efficiently checked.

Considered originally by Dolev and Yao (1983).

Improved algorithms by Dolev, Even and Karp.

Introduces some of the techniques required for more complex protocols.

Ping-Pong protocols consist of a sequence of message exchanges between a pair of participants. A sequence of operators is applied on a received message to compute the message sent.

$$X \to Y\text{:} \quad \{M\}_{K_Y}, X$$

$$Y \to X\text{:} \quad \{M\}_{K_X}$$

X sends a secret message M to Y who responds to confirm the reception of the secret message.

Does the message M remain secret ?

$$X \to Y\text{:} \quad \{M\}_{K_Y}, X$$
$$Y \to X\text{:} \quad \{M\}_{K_X}$$

X sends a secret message M to Y who responds to confirm the reception of the secret message.

Does the message M remain secret ?

An attack:

$$\begin{array}{ll} X \to (Y)Z & : \{M\}_{K_Y}, X \\ Z \to Y & : \{M\}_{K_Y}, Z \\ Y \to Z & : \{M\}_{K_Z} \end{array}$$

Attacker Z computes M from the last message.

$$X \to Y\text{:}\quad \{M\}_{K_Y}, X$$
$$Y \to X\text{:}\quad \{M\}_{K_X}$$

To correct the flaw the sender's identity is now sent encrypted.

$$X \to Y\text{:}\quad \{M, X\}_{K_Y}$$
$$Y \to X\text{:}\quad \{M\}_{K_X}$$

$$X \to Y: \quad \{M\}_{K_Y}, X$$
$$Y \to X: \quad \{M\}_{K_X}$$

To correct the flaw the sender's identity is now sent encrypted.

$$X \to Y: \quad \{M, X\}_{K_Y}$$
$$Y \to X: \quad \{M\}_{K_X}$$

This protocol is secure.

$$X \rightarrow Y\text{:} \quad \{M\}_{K_Y}, X$$

$$Y \rightarrow X\text{:} \quad \{M\}_{K_X}$$

To correct the flaw the sender's identity is now sent encrypted.

$$X \rightarrow Y\text{:} \quad \{M, X\}_{K_Y}$$

$$Y \rightarrow X\text{:} \quad \{M\}_{K_X}$$

This protocol is secure. Proof ?

$$X \to Y\text{:} \quad \{M\}_{K_Y}, X$$
$$Y \to X\text{:} \quad \{M\}_{K_X}$$

To correct the flaw the sender's identity is now sent encrypted.

$$X \to Y\text{:} \quad \{M, X\}_{K_Y}$$
$$Y \to X\text{:} \quad \{M\}_{K_X}$$

This protocol is secure. Proof ?

To make it more secure:

$$X \to Y\text{:} \quad \{\{M\}_{K_Y}, X\}_{K_Y}$$
$$Y \to X\text{:} \quad \{M\}_{K_X}$$

$$X \rightarrow Y\text{:} \quad \{M\}_{K_Y}, X$$
$$Y \rightarrow X\text{:} \quad \{M\}_{K_X}$$

To correct the flaw the sender's identity is now sent encrypted.

$$X \rightarrow Y\text{:} \quad \{M, X\}_{K_Y}$$
$$Y \rightarrow X\text{:} \quad \{M\}_{K_X}$$

This protocol is secure. Proof ?

To make it more secure:

$$X \rightarrow Y\text{:} \quad \{\{M\}_{K_Y}, X\}_{K_Y}$$
$$Y \rightarrow X\text{:} \quad \{M\}_{K_X}$$

This protocol is insecure!!

$$X \rightarrow Y: \quad \{\{M\}_{K_Y}, X\}_{K_Y}$$

$$Y \rightarrow X: \quad \{M\}_{K_X}$$

Attack:

$$\begin{aligned}
X &\rightarrow (Y)Z &&: \{\{M\}_{K_Y}, X\}_{K_Y} \\
Z &\rightarrow Y &&: \{\{\{M\}_{K_Y}, X\}_{K_Y}, Z\}_{K_Y} \\
Y &\rightarrow Z &&: \{\{M\}_{K_Y}, X\}_{K_Z} \\
Z &\rightarrow Y &&: \{\{M\}_{K_Y}, Z\}_{K_Y} \\
Y &\rightarrow Z &&: \{M\}_{K_Z}
\end{aligned}$$

$$X \rightarrow Y\text{:} \quad \{\{M\}_{K_Y}, X\}_{K_Y}$$

$$Y \rightarrow X\text{:} \quad \{M\}_{K_X}$$

Attack:

$$\begin{aligned}
X &\rightarrow (Y)Z &&: \{\{M\}_{K_Y}, X\}_{K_Y} \\
Z &\rightarrow Y &&: \{\{\{M\}_{K_Y}, X\}_{K_Y}, Z\}_{K_Y} \\
Y &\rightarrow Z &&: \{\{M\}_{K_Y}, X\}_{K_Z} \\
Z &\rightarrow Y &&: \{\{M\}_{K_Y}, Z\}_{K_Y} \\
Y &\rightarrow Z &&: \{M\}_{K_Z}
\end{aligned}$$

Another attack: Z reads $\{M\}_{K_X}$, then

$$\begin{aligned}
Z &\rightarrow X &&: \{\{M\}_{K_X}, Z\}_{K_X} \\
X &\rightarrow Z &&: \{M\}_{K_Z}
\end{aligned}$$

Each user X has a public key K_X and private key K_X^{-1}. We have operators

$$\begin{aligned} E_X(m) &= \{m\}_{K_X} \\ i_X(m) &= m, X \end{aligned}$$

as well as their inverse operations. The protocol $P(X,Y)$

$$\begin{aligned} X \to Y\text{:} &\quad \{M\}_{K_Y}, X \\ Y \to X\text{:} &\quad \{M\}_{K_X} \end{aligned}$$

is denoted by the rules:

send($i_X(E_Y(M))$)
receive($i_X(E_Y(x))$), send($E_X(x)$)

Our protocols are sets of receive-send pairs. We follow the usual Dolev-Yao model: communication takes place through the intruder.

$$X \rightarrow Y\text{:} \quad \{M, X\}_{K_Y}$$
$$Y \rightarrow X\text{:} \quad \{M\}_{K_X}$$

is modeled as

send$(E_Y(i_X(M)))$
receive$(E_Y(i_X(x)))$, send$(E_X(x))$

$$X \rightarrow Y\text{:} \quad \{\{M\}_{K_Y}, X\}_{K_Y}$$
$$Y \rightarrow X\text{:} \quad \{M\}_{K_X}$$

is modeled as

send$(E_Y(i_X(E_Y(M))))$
receive$(E_Y(i_X(E_Y(x))))$, send$(E_X(x))$

In general protocols may have more than two steps.

We are interested in modeling the knowledge of the intruder.

Consider two honest agents A, B and an attacker C (justification ??)

A rule like send$(E_Y(i_X(E_Y(M))))$ is expanded to six rules

send$(E_B(i_A(E_B(M_{AB}))))$ send$(E_C(i_A(E_C(M_{AC}))))$ ...

... or just one rule send$(E_B(i_A(E_B(M))))$ suffices!

Rule receive$(E_Y(i_X(E_Y(x))))$, send$(E_X(x))$ is expanded to six rules

receive$(E_B(i_A(E_B(x))))$, send$(E_A(x))$

...

These are essentially rules for modeling intruder's ability to learn new messages from existing messages.

intruder knows $E_B(i_A(E_B(M)))$

if intruder knows $E_B(i_A(E_B(x)))$ then intruder knows $E_A(x)$

These rules can now be thought of as rules for manipulating a stack.

stack $E_B(i_A(E_B(M)))$ is reachable
if stack $E_B(i_A(E_B(x)))$ is reachable then stack $E_A(x)$ is reachable.

Consider pop and push as basic operations.

$\rightarrow q_0(M)$	(push)	$q(E_B(x))\rightarrow q_3(x)$	(pop)
$q_0(x)\rightarrow q_1(E_B(x))$	(push)	$q_3(i_A(x))\rightarrow q_4(x)$	(pop)
$q_1(x)\rightarrow q_2(i_A(x))$	(push)	$q_4(E_B(x))\rightarrow q_5(x)$	(pop)
$q_2(x)\rightarrow q(E_B(x))$	(push)	$q_5(x)\rightarrow q(E_A(x))$	(push)

Besides, we have default rules: $q(x) \rightarrow q(E_C(x))$, $q(E_C(x)) \rightarrow q(x)$, $q(x) \rightarrow q(i_A(x))$, $q(i_A(x)) \rightarrow q(x)$,...

The insecurity question: can $q(M)$ be obtained from these rules?

To answer this, we add new derived rules. For any p, p', p'':

given push rule $p(x) \longrightarrow p'(\sigma(x))$ and pop rule $p'(\sigma(x)) \longrightarrow p''(x)$
we add the ϵ-rule $p(x) \longrightarrow p''(x)$ $\qquad$ ($\sigma \in \{E_A, i_A, \ldots\}$).

given push rule $p(x) \longrightarrow p'(\sigma(x))$ and ϵ-rule $p'(x) \longrightarrow p''(x)$
we add the push rule $p(x) \longrightarrow p''(\sigma(x))$

given push rule $p'(M)$ and ϵ-rule $p'(x) \longrightarrow p''(x)$
we add the push rule $p''(M)$

In this way, if the rule $q(M)$ is eventually derived
then the protocol is insecure, otherwise the protocol is secure.

Correctness argument

Claim: if any $p(t)$ can be obtained, then it can be obtained using only the (old and new) push rules.

Induction on the number of rules applied to obtain $p(t)$.

If $p(M)$ is obtained by the same rule then there is nothing to show.

If $p(\sigma(t'))$ is obtained by applying push rule $p'(x) \rightarrow p(\sigma(x))$ on $p'(t')$, by induction hypothesis, $p'(t')$ can be obtained using only push rules, hence so can be $p(\sigma(t'))$.

Let $p(t)$ be obtained by applying ϵ-rule $p'(x) \longrightarrow p(x)$ on $p'(t)$.

By i.h. $p'(t)$ can be obtained using only push rules.

If $t = M$ and $p'(M)$ is obtained using the same rule, then the derived rule $p(M)$ does the job.

If $t = \sigma(t')$ and $p'(\sigma(t'))$ is obtained by applying push rule $p''(x) \longrightarrow p'(\sigma(x))$ on $p''(t')$, then the derived rule $p''(x) \longrightarrow p(\sigma(x))$ does the job.

Let $p(t)$ be obtained from $p'(\sigma(t'))$ by applying pop rule $p'(\sigma(x)) \longrightarrow p(x)$ on $p'(\sigma(x))$.

By i.h. $p'(\sigma(t))$ can be obtained using only push rules.

$p'(\sigma(t))$ must be obtained by applying a push rule $p''(x) \longrightarrow p'(\sigma(x))$ on $p''(t)$.

We have a derived rule $p''(x) \longrightarrow p(x)$.

If $t = M$ and $p''(M)$ is obtained from the same rule then the rule $p(M)$ does the job

If $t = \sigma'(t')$ and $p''(\sigma'(t'))$ is obtained by applying push rule $p'''(x) \longrightarrow p''(\sigma'(x))$ on $p'''(t'')$ then the derived rule $p'''(x) \longrightarrow p(\sigma'(x))$ does the job.

We have shown:

Claim: if any $p(t)$ can be obtained, then it can be obtained using only the (old and new) push rules.

Now suppose the protocol is insecure.

Then $q(M)$ must be obtained by the rules.

By above claim, it must be obtained by only the push rules.

The only possibility is that it is obtained by the rule $q(M)$.

Hence the rule $q(M)$ must be derived.

Time complexity

Number of states s is linear in the size of the input protocol.

For a fixed set of operators, the number of possible rules is $O(s^2)$.

Our algorithm runs in a loop, adding as many new derived rules as possible, till no further rules can be added.

A loose analysis shows us that this is a polynomial time algorithm.

Finer analysis actually gives a cubic time complexity.

$$X \to Y\text{:} \quad \{M\}_{K_Y}, X$$

$$Y \to X\text{:} \quad \{M\}_{K_X}$$

$X \rightarrow Y$: $\{M\}_{K_Y}, X$

$Y \rightarrow X$: $\{M\}_{K_X}$

Some useful rules:

$$\rightarrow q_0(M)$$

$$q_0(x) \rightarrow q_1(E_B(x))$$

$$q_1(x) \rightarrow q(i_A(x))$$

$$q(i_C(x)) \rightarrow q_2(x)$$

$$q_2(E_B(x)) \rightarrow q_3(x)$$

$$q_3(x) \rightarrow q(E_C(x))$$

$$q(i_A(x)) \rightarrow q(x)$$

$$q(x) \rightarrow q(i_C(x))$$

$$q(E_C(x)) \rightarrow q(x)$$

$X \rightarrow Y$: $\{M\}_{K_Y}, X$

$Y \rightarrow X$: $\{M\}_{K_X}$

Some useful rules:

$$\rightarrow q_0(M)$$

$$q_0(x) \rightarrow q_1(E_B(x))$$

$$q_1(x) \rightarrow q(i_A(x))$$

$$q(i_C(x)) \rightarrow q_2(x)$$

$$q_2(E_B(x)) \rightarrow q_3(x)$$

$$q_3(x) \rightarrow q(E_C(x))$$

$$q(i_A(x)) \rightarrow q(x)$$

$$q(x) \rightarrow q(i_C(x))$$

$$q(E_C(x)) \rightarrow q(x)$$

Derived rules:

$$q_1(x) \quad \rightarrow q(x)$$

$X \to Y$: $\{M\}_{K_Y}, X$

$Y \to X$: $\{M\}_{K_X}$

Some useful rules:

$$\to q_0(M)$$

$$q_0(x) \to q_1(E_B(x))$$

$$q_1(x) \to q(i_A(x))$$

$$q(i_C(x)) \to q_2(x)$$

$$q_2(E_B(x)) \to q_3(x)$$

$$q_3(x) \to q(E_C(x))$$

$$q(i_A(x)) \to q(x)$$

$$q(x) \to q(i_C(x))$$

$$q(E_C(x)) \to q(x)$$

Derived rules:

$$q_1(x) \quad \to q(x)$$

$$q_0(x) \quad \to q(E_B(x))$$

$X \to Y$: $\{M\}_{K_Y}, X$

$Y \to X$: $\{M\}_{K_X}$

Some useful rules:

$$\to q_0(M)$$

$$q_0(x) \to q_1(E_B(x))$$

$$q_1(x) \to q(i_A(x))$$

$$q(i_C(x)) \to q_2(x)$$

$$q_2(E_B(x)) \to q_3(x)$$

$$q_3(x) \to q(E_C(x))$$

$$q(i_A(x)) \to q(x)$$

$$q(x) \to q(i_C(x))$$

$$q(E_C(x)) \to q(x)$$

Derived rules:

$$q_1(x) \quad \to q(x)$$

$$q_0(x) \quad \to q(E_B(x))$$

$$q(x) \quad \to q_2(x)$$

$X \rightarrow Y$: $\{M\}_{K_Y}, X$

$Y \rightarrow X$: $\{M\}_{K_X}$

Some useful rules:

$$\rightarrow q_0(M)$$

$$q_0(x) \rightarrow q_1(E_B(x))$$

$$q_1(x) \rightarrow q(i_A(x))$$

$$q(i_C(x)) \rightarrow q_2(x)$$

$$q_2(E_B(x)) \rightarrow q_3(x)$$

$$q_3(x) \rightarrow q(E_C(x))$$

$$q(i_A(x)) \rightarrow q(x)$$

$$q(x) \rightarrow q(i_C(x))$$

$$q(E_C(x)) \rightarrow q(x)$$

Derived rules:

$$q_1(x) \quad \rightarrow \quad q(x)$$

$$q_0(x) \quad \rightarrow \quad q(E_B(x))$$

$$q(x) \quad \rightarrow \quad q_2(x)$$

$$q_0(x) \quad \rightarrow \quad q_2(E_B(x))$$

$X \to Y$: $\{M\}_{K_Y}, X$

$Y \to X$: $\{M\}_{K_X}$

Some useful rules:

$$\to q_0(M)$$

$$q_0(x) \to q_1(E_B(x))$$

$$q_1(x) \to q(i_A(x))$$

$$q(i_C(x)) \to q_2(x)$$

$$q_2(E_B(x)) \to q_3(x)$$

$$q_3(x) \to q(E_C(x))$$

$$q(i_A(x)) \to q(x)$$

$$q(x) \to q(i_C(x))$$

$$q(E_C(x)) \to q(x)$$

Derived rules:

$$q_1(x) \to q(x)$$

$$q_0(x) \to q(E_B(x))$$

$$q(x) \to q_2(x)$$

$$q_0(x) \to q_2(E_B(x))$$

$$q_0(x) \to q_3(x)$$

$X \to Y$: $\{M\}_{K_Y}, X$

$Y \to X$: $\{M\}_{K_X}$

Some useful rules:

$$\to q_0(M)$$

$$q_0(x) \to q_1(E_B(x))$$

$$q_1(x) \to q(i_A(x))$$

$$q(i_C(x)) \to q_2(x)$$

$$q_2(E_B(x)) \to q_3(x)$$

$$q_3(x) \to q(E_C(x))$$

$$q(i_A(x)) \to q(x)$$

$$q(x) \to q(i_C(x))$$

$$q(E_C(x)) \to q(x)$$

Derived rules:

$$q_1(x) \to q(x)$$

$$q_0(x) \to q(E_B(x))$$

$$q(x) \to q_2(x)$$

$$q_0(x) \to q_2(E_B(x))$$

$$q_0(x) \to q_3(x)$$

$$\to q_3(M)$$

$X \to Y$: $\{M\}_{K_Y}, X$

$Y \to X$: $\{M\}_{K_X}$

Some useful rules:

$$\to q_0(M)$$

$$q_0(x) \to q_1(E_B(x))$$

$$q_1(x) \to q(i_A(x))$$

$$q(i_C(x)) \to q_2(x)$$

$$q_2(E_B(x)) \to q_3(x)$$

$$q_3(x) \to q(E_C(x))$$

$$q(i_A(x)) \to q(x)$$

$$q(x) \to q(i_C(x))$$

$$q(E_C(x)) \to q(x)$$

Derived rules:

$$q_1(x) \to q(x)$$

$$q_0(x) \to q(E_B(x))$$

$$q(x) \to q_2(x)$$

$$q_0(x) \to q_2(E_B(x))$$

$$q_0(x) \to q_3(x)$$

$$\to q_3(M)$$

$$q_3(x) \to q(x)$$

$X \to Y$: $\{M\}_{K_Y}, X$

$Y \to X$: $\{M\}_{K_X}$

Some useful rules:

$$\to q_0(M)$$
$$q_0(x) \to q_1(E_B(x))$$
$$q_1(x) \to q(i_A(x))$$
$$q(i_C(x)) \to q_2(x)$$
$$q_2(E_B(x)) \to q_3(x)$$
$$q_3(x) \to q(E_C(x))$$
$$q(i_A(x)) \to q(x)$$
$$q(x) \to q(i_C(x))$$
$$q(E_C(x)) \to q(x)$$

Derived rules:

$$q_1(x) \to q(x)$$
$$q_0(x) \to q(E_B(x))$$
$$q(x) \to q_2(x)$$
$$q_0(x) \to q_2(E_B(x))$$
$$q_0(x) \to q_3(x)$$

$$\to q_3(M)$$
$$q_3(x) \to q(x)$$
$$\to q(M)$$

Insecure!