Rewriting in Protocol Verification

Stéphanie Delaune

Univ Rennes, CNRS, IRISA, France

Monday, June 29th, 2020

Cryptographic protocols everywhere !

Cryptographic protocols

- small programs designed to secure communication (*e.g.* secrecy, authentication, anonymity, . . .)
- use cryptographic primitives (*e.g.* encryption, signature, ...)

The network is unsecure!
Communications take place over a public network like the Internet.

Cryptographic protocols everywhere !

Cryptographic protocols

- small programs designed to secure communication (*e.g.* secrecy, authentication, anonymity, . . .)
- use cryptographic primitives (*e.g.* encryption, signature, ...)

It becomes more and more important to protect our privacy.

How cryptographic protocols can be attacked?

Cryptanalysis

- Differential attacks,
- Boomerang attacks,
- Cube attacks,
- ...

How cryptographic protocols can be attacked?

Logical attacks

- can be mounted even assuming perfect cryptography,
  $\hookrightarrow$ replay attack, man-in-the middle attack, ...
- subtle and hard to detect by "eyeballing" the protocol

This is the so-called **Dolev-Yao attacker** !

How cryptographic protocols can be attacked?

Logical attacks

- can be mounted even assuming perfect cryptography,
  $\hookrightarrow$ replay attack, man-in-the middle attack, ...
- subtle and hard to detect by "eyeballing" the protocol

Example: An authentication flaw on the Needham Schroeder protocol

$$
\begin{aligned}
A \rightarrow B &: \{A, N_A\}_{\mathsf{pub}(B)} \\
B \rightarrow A &: \{N_A, N_B\}_{\mathsf{pub}(A)} \\
A \rightarrow B &: \{N_B\}_{\mathsf{pub}(B)}
\end{aligned}
$$

NS protocol (1978)

How cryptographic protocols can be attacked?

Logical attacks

- can be mounted even assuming perfect cryptography,
  $\hookrightarrow$ replay attack, man-in-the middle attack, ...
- subtle and hard to detect by "eyeballing" the protocol

Example: An authentication flaw on the Needham Schroeder protocol

$$\begin{aligned} A &\rightarrow B : \{A, N_A\}_{\mathsf{pub}(B)} \\ B &\rightarrow A : \{N_A, N_B\}_{\mathsf{pub}(A)} \\ A &\rightarrow B : \{N_B\}_{\mathsf{pub}(B)} \end{aligned}$$

NS protocol (1978)

$$\begin{aligned} A &\rightarrow B : \{A, N_A\}_{\mathsf{pub}(B)} \\ B &\rightarrow A : \{N_A, N_B, B\}_{\mathsf{pub}(A)} \\ A &\rightarrow B : \{N_B\}_{\mathsf{pub}(B)} \end{aligned}$$

NS-Lowe protocol (1995)

How cryptographic protocols can be attacked?

Logical attacks

- can be mounted even assuming perfect cryptography,
  $\hookrightarrow$ replay attack, man-in-the middle attack, ...
- subtle and hard to detect by "eyeballing" the protocol

Example: FREAK attack by Barghavan et al. (2015)

A logical flaw that allows a *man-in-the-middle* attacker to downgrade connections from 'strong' RSA to 'export grade' RSA.

How cryptographic protocols can be attacked?

Logical attacks

- can be mounted even assuming perfect cryptography,
  $\hookrightarrow$ replay attack, man-in-the middle attack, ...
- subtle and hard to detect by "eyeballing" the protocol

Example: A traceability attack on the BAC protocol (2010)

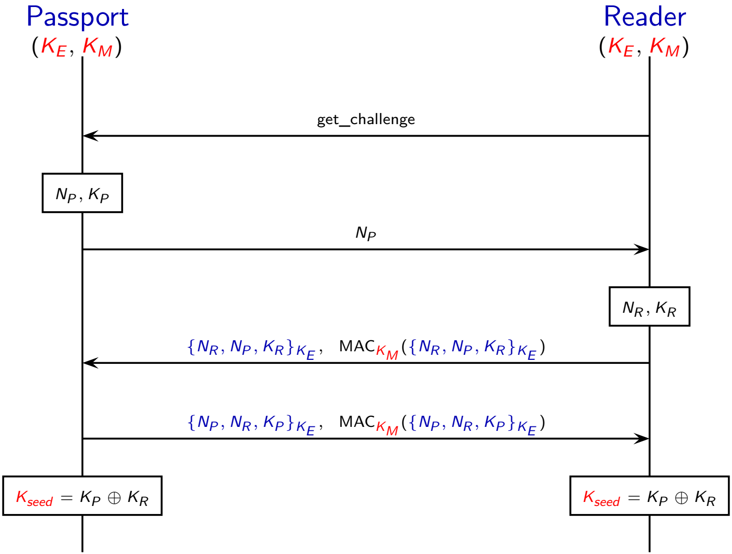
Security

Defects in e-passports allow real-time tracking

This threat brought to you by RFID

The register - Jan. 2010

Basic Acccess Control (BAC) protocol



Unlinkability/Untraceability

Informally, an observer/attacker can not observe the difference between the two following situations:

1. a situation where the same passport may be used twice (or even more);
2. a situation where each passport is used at most once.

Unlinkability/Untraceability

Informally, an observer/attacker can not observe the difference between the two following situations:

1. a situation where the same passport may be used twice (or even more);
2. a situation where each passport is used at most once.

More formally,

$$!\text{new } ke.\text{new } km.(!P_{\mathsf{BAC}} \mid !R_{\mathsf{BAC}}) \overset{?}{\approx} !\text{new } ke.\text{new } km.(P_{\mathsf{BAC}} \mid R_{\mathsf{BAC}})$$

many sessions for each passport

only one session for each passport

(we still have to formalize the notion of equivalence)

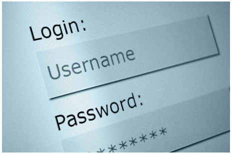
Some other equivalence-based security properties

Vote privacy

the fact that a particular voter voted in a particular way is not revealed to anyone

Strong secrecy

the fact that an adversary cannot see any difference when the value of the secret changes
$\longrightarrow$ stronger than the notion of secrecy as non-deducibility.



Guessing attack

the fact that an adversary can not learn the value of passwords even if he knows that they have been choosen in a particular dictionary.

How rewriting and unification theory

can help us

in protocol verification?

Messages as terms - Back to the BAC protocol

Nonces n_r, n_p, and keys k_r, k_p, k_e, k_m are modelled using names

Cryptographic primitives are modelled using function symbols

- encryption/decryption: senc/2, sdec/2
- concatenation/projections: $\langle\, ,\, \rangle/2$, $\mathsf{proj}_1/1$, $\mathsf{proj}_2/1$
- mac construction: mac/2

$$\mathsf{sdec}(\mathsf{senc}(x,y),y) = x \quad \mathsf{proj}_1(\langle x,y\rangle) = x \quad \mathsf{proj}_2(\langle x,y\rangle) = y$$

Messages as terms - Back to the BAC protocol

Nonces n_r, n_p, and keys k_r, k_p, k_e, k_m are modelled using names

Cryptographic primitives are modelled using function symbols

- encryption/decryption: senc/2, sdec/2
- concatenation/projections: $\langle\ ,\ \rangle/2$, $\mathsf{proj}_1/1$, $\mathsf{proj}_2/1$
- mac construction: mac/2

$$\mathsf{sdec}(\mathsf{senc}(x,y),y) = x \quad \mathsf{proj}_1(\langle x,y\rangle) = x \quad \mathsf{proj}_2(\langle x,y\rangle) = y$$

Exclusive-or operator: $\oplus$ of arity 2 and 0 (neutral element)

$$\begin{array}{rclcrcl} x \oplus (y \oplus z) & = & (x \oplus y) \oplus z & \quad & x \oplus x & = & 0 \\ x \oplus y & = & y \oplus x & & x \oplus 0 & = & x \end{array}$$

Messages as terms - Back to the BAC protocol

Nonces n_r, n_p, and keys k_r, k_p, k_e, k_m are modelled using names

Cryptographic primitives are modelled using function symbols

- encryption/decryption: senc/2, sdec/2
- concatenation/projections: $\langle\, ,\, \rangle/2$, $\mathsf{proj}_1/1$, $\mathsf{proj}_2/1$
- mac construction: mac/2

$$\mathsf{sdec}(\mathsf{senc}(x, y), y) = x \quad \mathsf{proj}_1(\langle x, y\rangle) = x \quad \mathsf{proj}_2(\langle x, y\rangle) = y$$

Equational theories are useful to model algebraic properties of cryptographic primitives.

Computations as recipes

frame = knowledge of the attacker = sequence of messages

$$\phi = \{w_1 \triangleright u_1, \ldots, w_\ell \triangleright u_\ell\}$$

Computations as recipes

frame = knowledge of the attacker = sequence of messages

$$\phi = \{w_1 \triangleright u_1, \ldots, w_\ell \triangleright u_\ell\}$$

Example: $\mathsf{adec}(\mathsf{aenc}(x, \mathsf{pk}(y)), y) \to x$

$$\{\underbrace{w_1 \triangleright \mathsf{pk}(ska);\ w_2 \triangleright \mathsf{pk}(skb);\ w_3 \triangleright skc;}_{\text{initial knowledge}}\quad \underbrace{w_4 \triangleright \mathsf{aenc}(\langle a, n_a\rangle, \mathsf{pk}(skc))}_{\text{1st message of NS}}\}.$$

Computations as recipes

frame $=$ knowledge of the attacker $=$ sequence of messages

$$\phi = \{w_1 \triangleright u_1, \ldots, w_\ell \triangleright u_\ell\}$$

Example: $\mathsf{adec}(\mathsf{aenc}(x, \mathsf{pk}(y)), y) \to x$

$$\underbrace{\{w_1 \triangleright \mathsf{pk}(ska);\ w_2 \triangleright \mathsf{pk}(skb);\ w_3 \triangleright skc;}_{\text{initial knowledge}} \quad \underbrace{w_4 \triangleright \mathsf{aenc}(\langle a, n_a\rangle, \mathsf{pk}(skc))}_{\text{1st message of NS}}\}.$$

Some recipes:

- from his private key skc, the attacker is able to get his public key with $R = \mathsf{pk}(w_3)$;
- $R = \mathsf{aenc}(\mathsf{adec}(w_4, w_3), w_2)$ – this is the first step of the man-in-the-middle attack on NS protocol

Computations as recipes

frame = knowledge of the attacker = sequence of messages

$$\phi = \{w_1 \triangleright u_1, \ldots, w_\ell \triangleright u_\ell\}$$

Example: $\mathsf{adec}(\mathsf{aenc}(x, \mathsf{pk}(y)), y) \to x$

$$\{\underbrace{w_1 \triangleright \mathsf{pk}(ska);\ w_2 \triangleright \mathsf{pk}(skb);\ w_3 \triangleright skc;}_{\text{initial knowledge}}\quad \underbrace{w_4 \triangleright \mathsf{aenc}(\langle a, n_a\rangle, \mathsf{pk}(skc))}_{\text{1st message of NS}}\}.$$

Some recipes:

- from his private key skc, the attacker is able to get his public key with $R = \mathsf{pk}(w_3)$;
- $R = \mathsf{aenc}(\mathsf{adec}(w_4, w_3), w_2)$ – this is the first step of the man-in-the-middle attack on NS protocol

Rewriting is useful to express computations performed by the attacker.

Static equivalence

Warm-up

$\longrightarrow$ this is the so-called passive attacker

The static equivalence problem ($\phi \sim \psi$)

- Input: two substitutions (called frames) ϕ and ψ

  $\phi = \{w_1 \triangleright u_1, \ldots, w_\ell \triangleright u_\ell\} \qquad \psi = \{w_1 \triangleright v_1, \ldots, w_\ell \triangleright v_\ell\}$
- Output: Can the attacker distinguish the two frames, *i.e.* does there exist a test $R_1 \stackrel{?}{=} R_2$ such that:

  $$R_1\phi =_E R_2\phi \text{ but } R_1\psi \neq_E R_2\psi \text{ (or the converse).}$$

The static equivalence problem ($\phi \sim \psi$)

- Input: two substitutions (called frames) ϕ and ψ

  $$\phi = \{w_1 \triangleright u_1, \ldots, w_\ell \triangleright u_\ell\} \qquad \psi = \{w_1 \triangleright v_1, \ldots, w_\ell \triangleright v_\ell\}$$

- Output: Can the attacker distinguish the two frames, *i.e.* does there exist a test $R_1 \stackrel{?}{=} R_2$ such that:

  $$R_1\phi =_E R_2\phi \text{ but } R_1\psi \neq_E R_2\psi \text{ (or the converse).}$$

Example 1: $\mathsf{adec}(\mathsf{aenc}(x, \mathsf{pk}(y)), y) = x$

- $\phi = \{w_1 \triangleright \mathsf{pk}(sks);\ w_2 \triangleright \mathsf{aenc}(yes, \mathsf{pk}(sks))\}$; and
- $\psi = \{w_1 \triangleright \mathsf{pk}(sks);\ w_2 \triangleright \mathsf{aenc}(no, \mathsf{pk}(sks))\}$.

The static equivalence problem ($\phi \sim \psi$)

- Input: two substitutions (called frames) ϕ and ψ

$$\phi = \{w_1 \triangleright u_1, \ldots, w_\ell \triangleright u_\ell\} \qquad \psi = \{w_1 \triangleright v_1, \ldots, w_\ell \triangleright v_\ell\}$$

- Output: Can the attacker distinguish the two frames, *i.e.* does there exist a test $R_1 \stackrel{?}{=} R_2$ such that:

$$R_1\phi =_E R_2\phi \text{ but } R_1\psi \neq_E R_2\psi \text{ (or the converse).}$$

Example 1: $\mathrm{adec}(\mathrm{aenc}(x, \mathrm{pk}(y)), y) = x$

- $\phi = \{w_1 \triangleright \mathrm{pk}(sks);\ w_2 \triangleright \mathrm{aenc}(yes, \mathrm{pk}(sks))\}$; and
- $\psi = \{w_1 \triangleright \mathrm{pk}(sks);\ w_2 \triangleright \mathrm{aenc}(no, \mathrm{pk}(sks))\}$.

$\longrightarrow$ They are not in static equivalence: $\mathrm{aenc}(yes, w_1) \stackrel{?}{=} w_2$.

The static equivalence problem ($\phi \sim \psi$)

- Input: two substitutions (called frames) ϕ and ψ

$$\phi = \{w_1 \triangleright u_1, \ldots, w_\ell \triangleright u_\ell\} \qquad \psi = \{w_1 \triangleright v_1, \ldots, w_\ell \triangleright v_\ell\}$$

- Output: Can the attacker distinguish the two frames, *i.e.* does there exist a test $R_1 \stackrel{?}{=} R_2$ such that:

$$R_1\phi =_E R_2\phi \text{ but } R_1\psi \neq_E R_2\psi \text{ (or the converse).}$$

Example 2: (randomized encryption)

- $\phi = \{w_1 \triangleright \mathsf{pk}(sks);\ w_2 \triangleright \mathsf{aenc}(\langle yes, r\rangle, \mathsf{pk}(sks))\}$; and
- $\psi = \{w_1 \triangleright \mathsf{pk}(sks);\ w_2 \triangleright \mathsf{aenc}(\langle no, r\rangle, \mathsf{pk}(sks))\}$.

The static equivalence problem ($\phi \sim \psi$)

- Input: two substitutions (called frames) ϕ and ψ

  $\phi = \{w_1 \triangleright u_1, \ldots, w_\ell \triangleright u_\ell\} \qquad \psi = \{w_1 \triangleright v_1, \ldots, w_\ell \triangleright v_\ell\}$
- Output: Can the attacker distinguish the two frames, *i.e.* does there exist a test $R_1 \stackrel{?}{=} R_2$ such that:

  $$R_1\phi =_E R_2\phi \text{ but } R_1\psi \neq_E R_2\psi \text{ (or the converse).}$$

Example 2: (randomized encryption)

- $\phi = \{w_1 \triangleright \mathsf{pk}(sks);\ w_2 \triangleright \mathsf{aenc}(\langle yes, r\rangle, \mathsf{pk}(sks))\}$; and
- $\psi = \{w_1 \triangleright \mathsf{pk}(sks);\ w_2 \triangleright \mathsf{aenc}(\langle no, r\rangle, \mathsf{pk}(sks))\}$.

$\longrightarrow$ They are in static equivalence.

Static equivalence – some existing results

Theory E	Deduction	**Static Equivalence**
subterm convergent	PTIME	
blind signature, homo. encryption	decidable [Abadi & Cortier, 06]	
ACU	NP-complete	PTIME [Cortier & D., 10]
ACUN/AG	PTIME [Chevalier et al, 03]	PTIME [Cortier & D., 10]
ACUNh/AGh	PTIME [D., 06]	decidable [Cortier & D., 10]

The case of monoidal theories, e.g. XOR, AG, ...

Getting some inspiration from existing results and proofs in unification theory, e.g. [Nutt, 90], and [Baader & Schulz, 96], we associate a semi-ring S_{E} to a monoidal theory E:

$$\mathbb{Z}/2\mathbb{Z} \text{ for ACUN}, \quad \mathbb{Z} \text{ for AG}, \quad \mathbb{Z}/2\mathbb{Z}[\mathsf{X}] \text{ for ACUNh}, \ \ldots$$

The case of monoidal theories, e.g. XOR, AG, ...

Getting some inspiration from existing results and proofs in unification theory, e.g. [Nutt, 90], and [Baader & Schulz, 96], we associate a semi-ring S_{E} to a monoidal theory E:

$\mathbb{Z}/2\mathbb{Z}$ for ACUN, $\mathbb{Z}$ for AG, $\mathbb{Z}/2\mathbb{Z}[\mathrm{X}]$ for ACUNh, ...

Theorem [Cortier & D., 10]

Static equivalence in E is reducible in PTIME to

Input Two matrices A_1 and A_2 over S_E

Output Does the following equality holds?

$$\{(X,Y) \in S_E^{\ell} \times S_E^{\ell} \mid X \cdot A_1 = Y \cdot A_1\}$$
$$=$$
$$\{(X,Y) \in S_E^{\ell} \times S_E^{\ell} \mid X \cdot A_2 = Y \cdot A_2\}$$

Reduction on an example – $\mathsf{E} = \mathsf{AG}$ and $S_{\mathsf{E}} = \mathbb{Z}$

$\longrightarrow$ a and b two constants (e.g. nonces)

- $\phi_1 = \{w_1 \triangleright a + b + b;\ w_2 \triangleright b;\ w_3 \triangleright a + a + a\}$

Reduction on an example – $\mathsf{E} = \mathsf{AG}$ and $S_{\mathsf{E}} = \mathbb{Z}$

$\longrightarrow$ a and b two constants (e.g. nonces)

- $\phi_1 = \{w_1 \triangleright a + b + b;\ w_2 \triangleright b;\ w_3 \triangleright a + a + a\}$

Question: Is a deducible from ϕ_1, i.e does there exists X such that $X \cdot A_1 = V$?

$$A_1 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 3 & 0 \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Reduction on an example – E = AG and $S_{\mathsf{E}} = \mathbb{Z}$

$\longrightarrow$ a and b two constants (e.g. nonces)

- $\phi_1 = \{w_1 \triangleright a + b + b;\ w_2 \triangleright b;\ w_3 \triangleright a + a + a\}$

Question: Is a deducible from ϕ_1, i.e does there exists X such that $X \cdot A_1 = V$?

$$A_1 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 3 & 0 \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Answer: Yes ! $X = (1\ \ -2\ \ 0)$.

$\longrightarrow$ this corresponds to recipe $R = w_1 + -(w_2) + -(w_2)$.
Indeed, we have that:

$$\begin{aligned} R\phi_1\downarrow &= a + b + b + -(b) + -(b) \\ &= a \end{aligned}$$

Reduction on an example – $\mathsf{E} = \mathsf{AG}$ and $S_{\mathsf{E}} = \mathbb{Z}$

$\longrightarrow$ a and b two constants (e.g. nonces)

- $\phi_1 = \{w_1 \triangleright a + b + b;\ w_2 \triangleright b;\ w_3 \triangleright a + a + a\}$
- $\phi_2 = \{w_1 \triangleright a + b + b;\ w_2 \triangleright b + b;\ w_3 \triangleright 3a - 6b\}$

Question: Is $\phi_1 \overset{?}{\sim} \phi_2$, i.e. do $\{(X, Y) \mid X \cdot A_1 = Y \cdot A_1\}$ and $\{(X, Y) \mid X \cdot A_2 = Y \cdot A_2\}$ have the same set of solutions?

$$A_1 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 3 & 0 \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} 1 & 2 \\ 0 & 2 \\ 3 & -6 \end{pmatrix}$$

Reduction on an example – $\mathsf{E} = \mathsf{AG}$ and $S_{\mathsf{E}} = \mathbb{Z}$

$\longrightarrow$ a and b two constants (e.g. nonces)

- $\phi_1 = \{w_1 \triangleright a + b + b;\ w_2 \triangleright b;\ w_3 \triangleright a + a + a\}$
- $\phi_2 = \{w_1 \triangleright a + b + b;\ w_2 \triangleright b + b;\ w_3 \triangleright 3a - 6b\}$

Question: Is $\phi_1 \overset{?}{\sim} \phi_2$, i.e. do $\{(X, Y) \mid X \cdot A_1 = Y \cdot A_1\}$ and $\{(X, Y) \mid X \cdot A_2 = Y \cdot A_2\}$ have the same set of solutions?

$$A_1 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 3 & 0 \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} 1 & 2 \\ 0 & 2 \\ 3 & -6 \end{pmatrix}$$

Answer: Yes ! $\phi_1 \sim_{\mathsf{E}} \phi_2$

Reduction on an example – $\mathsf{E} = \mathsf{AG}$ and $S_{\mathsf{E}} = \mathbb{Z}$

$\longrightarrow$ a and b two constants (e.g. nonces)

- $\phi_1 = \{w_1 \triangleright a + b + b;\ w_2 \triangleright b;\ w_3 \triangleright a + a + a\}$
- $\phi_2 = \{w_1 \triangleright a + b + b;\ w_2 \triangleright b + b;\ w_3 \triangleright 3a - 6b\}$

Question: Is $\phi_1 \overset{?}{\sim} \phi_2$, i.e. do $\{(X, Y) \mid X \cdot A_1 = Y \cdot A_1\}$ and $\{(X, Y) \mid X \cdot A_2 = Y \cdot A_2\}$ have the same set of solutions?

$$A_1 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 3 & 0 \end{pmatrix} \quad \text{and} \quad A_2 = \begin{pmatrix} 1 & 2 \\ 0 & 2 \\ 3 & -6 \end{pmatrix}$$

Answer: Yes ! $\phi_1 \sim_{\mathsf{E}} \phi_2$

In both cases (deduction and equivalence), we only have to consider sets of linear equations with coefficient in $\mathbb{Z}$.

Combination result

If deduction and static equivalence are decidable for two disjoint theories E_1 and E_2 then they are also decidable for $E_1 \cup E_2$.

[Cortier & D., 10]

Example: this allows one to combine encryption and exclusive-or

Combination result

If deduction and static equivalence are decidable for two disjoint theories E_1 and E_2 then they are also decidable for $\mathsf{E}_1 \cup \mathsf{E}_2$.

[Cortier & D., 10]

Example: this allows one to combine encryption and exclusive-or

Proof (sketch): given ϕ and ψ built on $E_1 \cup E_2$

1. saturate both frames simultaneously with deducible subterms (notion of alien subterms); $\longrightarrow$ this leads to ϕ_+ and ψ_+
2. abstract subterms coming from E_i ($i = 1, 2$) $\longrightarrow$ this leads to $\overline{\phi_+}^i$ and $\overline{\psi_+}^i$
3. check whether: $\overline{\phi_+}^i \approx_{E_{2-i}} \overline{\psi_+}^i$ (with $i = 1, 2$)

$\longrightarrow$ inspiration from [Schmidt-Schauß, 89; Baader & Schluz, 96]

Caution !

One should never underestimate the attacker !

The attacker can listen to the communication but also:

- intercept the messages that are sent by the participants,
- build new messages according to his deduction capabilities, and
- send messages on the communication network.

$\longrightarrow$ this is the co-called active attacker

from frames to constraint systems

A constraint system $\mathcal{C}$ is a triple $(\phi; \mathcal{D}; \mathcal{E})$ where:

1. $\phi = \{w_1 \rhd v_1;\ \ldots; w_\ell \rhd v_\ell\}$ is an open frame;
2. $\mathcal{D}$ is a set of deducibility constraints: $X \overset{?}{\rhd} x$ with $ar(X) < \ell$
3. $\mathcal{E}$ is a unification problem (modulo E)

+ some conditions (e.g. monotonicity, origination).

from frames to constraint systems

A constraint system $\mathcal{C}$ is a triple $(\phi; \mathcal{D}; \mathcal{E})$ where:

1. $\phi = \{w_1 \triangleright v_1;\ \ldots; w_\ell \triangleright v_\ell\}$ is an open frame;
2. $\mathcal{D}$ is a set of deducibility constraints: $X \overset{?}{\triangleright} x$ with $ar(X) < \ell$
3. $\mathcal{E}$ is a unification problem (modulo E)

$+$ some conditions (e.g. monotonicity, origination).

Example: a simple challenge-response protocol

$$\begin{array}{ll} A \rightarrow B : & \{\mathrm{req}, n\}_k \\ B \rightarrow A : & \{\mathrm{rep}, \mathrm{hash}(n)\}_k \end{array}$$

from frames to constraint systems

A constraint system $\mathcal{C}$ is a triple $(\phi; \mathcal{D}; \mathcal{E})$ where:

1. $\phi = \{w_1 \rhd v_1;\ \ldots; w_\ell \rhd v_\ell\}$ is an open frame;
2. $\mathcal{D}$ is a set of deducibility constraints: $X \overset{?}{\rhd} x$ with $ar(X) < \ell$
3. $\mathcal{E}$ is a unification problem (modulo E)

$+$ some conditions (e.g. monotonicity, origination).

Example: a simple challenge-response protocol

$$\begin{array}{rl} A \to B: & \{\mathsf{req}, n\}_k \\ B \to A: & \{\mathsf{rep}, \mathsf{hash}(n)\}_k \end{array}$$

1. $\{w_1 \rhd \{\langle \mathsf{req}, n\rangle\}_k;\ w_2 \rhd \{\langle \mathsf{rep}, \mathsf{hash}(\mathsf{proj}_2(\mathsf{sdec}(x, k)))\rangle\}_k\}$
2. $X \overset{?}{\rhd} x$ with $ar(X) = 1$
3. $\mathsf{proj}_1(\mathsf{sdec}(x, k)) = \mathsf{req}$

from frames to constraint systems

A constraint system $\mathcal{C}$ is a triple $(\phi; \mathcal{D}; \mathcal{E})$ where:

1. $\phi = \{w_1 \rhd v_1; \ \ldots; w_\ell \rhd v_\ell\}$ is an open frame;
2. $\mathcal{D}$ is a set of deducibility constraints: $X \overset{?}{\rhd} x$ with $ar(X) < \ell$
3. $\mathcal{E}$ is a unification problem (modulo E)

$+$ some conditions (e.g. monotonicity, origination).

Example: a simple challenge-response protocol

$$\begin{array}{ll} A \to B: & \{\mathsf{req}, n\}_k \\ B \to A: & \{\mathsf{rep}, \mathsf{hash}(n)\}_k \end{array}$$

1. $\{w_1 \rhd \{\langle \mathsf{req}, n\rangle\}_k; \ w_2 \rhd \{\langle \mathsf{rep}, \mathsf{hash}(\mathsf{proj}_2(\mathsf{sdec}(x, k)))\rangle\}_k\}$
2. $X \overset{?}{\rhd} x$ with $ar(X) = 1$
3. $\mathsf{proj}_1(\mathsf{sdec}(x, k)) = \mathsf{req}$

$\longrightarrow$ Solution: $X \mapsto w_1$ (and $x \mapsto \{\langle \mathsf{req}, n\rangle\}_k$).

Existing tools based on constraint solving

$\longrightarrow$ for a bounded number of sessions only

- **CL-AtSe** [Turuani, RTA'06], **OFMC** [Basin et al, 05] for reachability properties, e.g. secrecy, authentication: the security problem boils down to decide whether a constraint system admits a solution.

Existing tools based on constraint solving

$\longrightarrow$ for a bounded number of sessions only

- **CL-AtSe** [Turuani, RTA'06], **OFMC** [Basin et al, 05] for reachability properties, e.g. secrecy, authentication: the security problem boils down to decide whether a constraint system admits a solution.

- **DeepSec** [Cheval et al., 18] for equivalence-based properties, e.g. strong secrecy vote-privacy, unlinkability: the security problem boils down to decide whether two (sets) of constraint systems have the same set of solutions.

$\longrightarrow$ DeepSec also deals with $\neq$

DEEPSEC

Going back to monoidal theories

Remenber that we associate a semi-ring S_{E} to a monoidal theory E:

$\mathbb{Z}/2\mathbb{Z}$ for ACUN, $\mathbb{Z}$ for AG, $\mathbb{Z}/2\mathbb{Z}[\mathsf{X}]$ for ACUNh, ...

$\longrightarrow$ the previous encoding leads to quadratic equations.

Going back to monoidal theories

Remenber that we associate a semi-ring S_{E} to a monoidal theory E:

$\mathbb{Z}/2\mathbb{Z}$ for ACUN, $\mathbb{Z}$ for AG, $\mathbb{Z}/2\mathbb{Z}[\mathsf{X}]$ for ACUNh, ...

$\longrightarrow$ the previous encoding leads to quadratic equations.

However, it can be shown that they have a specific structure, and this has been exploited to derive the following results:

Theory E	S_{E}	Satisfiability	Equivalence
ACUN AG	$\mathbb{Z}/2\mathbb{Z}$ $\mathbb{Z}$	PTIME [Chevalier et al., 10]	PTIME [Delaune et al., 12]
ACUNh AGh	$\mathbb{Z}/2\mathbb{Z}[\mathsf{h}]$ $\mathbb{Z}[\mathsf{h}]$	PTIME [Delaune et al., 12]	PTIME [Delaune et al., 12]

A common difficulty in the active setting

Getting rid of rewriting steps that may occur inside $\mathcal{C} = (\phi; \mathcal{D}; \mathcal{E})$.

A common difficulty in the active setting

Getting rid of rewriting steps that may occur inside $\mathcal{C} = (\phi; \mathcal{D}; \mathcal{E})$.

Protocol:

$A \rightarrow B : \{n\}_k$
$B \rightarrow A : \{\mathsf{hash}(n)\}_k$

Constraint system $\mathcal{C} = (\phi; \mathcal{D}; \mathcal{E})$:

1. $w_1 \triangleright \{n\}_k;\ w_2 \triangleright \{\mathsf{hash}(\mathsf{sdec}(x, k))\}_k)$
2. $X \overset{?}{\triangleright} x$ with $ar(X) = 1$
3. $\mathcal{E} = \emptyset$.

A common difficulty in the active setting

Getting rid of rewriting steps that may occur inside $\mathcal{C} = (\phi; \mathcal{D}; \mathcal{E})$.

Protocol:

$A \to B: \ \{n\}_k$
$B \to A: \ \{\mathsf{hash}(n)\}_k$

Constraint system $\mathcal{C} = (\phi; \mathcal{D}; \mathcal{E})$:

1. $w_1 \triangleright \{n\}_k$; $\boxed{w_2 \triangleright \{\mathsf{hash}(\mathsf{sdec}(x, k))\}_k}$
2. $X \overset{?}{\triangleright} x$ with $ar(X) = 1$
3. $\mathcal{E} = \emptyset$.

Two kinds of solutions for $\mathcal{C}$:

- $\mathcal{C}_1$: either x is replaced by an encryption with k, i.e. $\{y\}_k$, and we can compute in advance that $\boxed{w_2 \triangleright \{\mathsf{hash}(y)\}_k}$;
- $\mathcal{C}_2$: or x is replaced by something else, and in this case $\boxed{w_2 \triangleright \{\mathsf{hash}(\mathsf{sdec}(x, k))\}_k}$.

No further rewriting step is then authorised in $\mathcal{C}_1$ and $\mathcal{C}_2$.

$\longrightarrow \mathsf{variant}(\mathcal{C}) = \{\mathcal{C}_1, \mathcal{C}_2\}$

Finite Variant Property (FVP) [Comon & D., 05]

An equational theory E (represented by a rewrite sytem $\downarrow$ - possibly modulo $\mathsf{E}' = \mathsf{AC}$) has the finite variant property if for any term t, we can compute a finite set of instances $t\sigma_1, \ldots, t\sigma_n$ such that:

$$\{t\sigma\downarrow \mid \sigma \in \Sigma\} = \bigcup_{i=1}^{n} \{t\sigma_i \downarrow \theta \mid \theta \in \Sigma\}$$

where Σ is the set of normalized substitutions.

$\longrightarrow$ variant$(t) = \{t\sigma_1, \ldots, t\sigma_n\}$.

Finite Variant Property (FVP) [Comon & D., 05]

An equational theory E (represented by a rewrite sytem $\downarrow$ - possibly modulo $\mathsf{E}' = \mathsf{AC}$) has the finite variant property if for any term t, we can compute a finite set of instances $t\sigma_1, \ldots, t\sigma_n$ such that:

$$\{t\sigma\downarrow \mid \sigma \in \Sigma\} = \bigcup_{i=1}^{n} \{t\sigma_i \downarrow \theta \mid \theta \in \Sigma\}$$

where Σ is the set of normalized substitutions.

$\longrightarrow$ variant$(t) = \{t\sigma_1, \ldots, t\sigma_n\}$.

Examples:

- Symmetric, asymmetric encryptions, (blind) signatures, several equational theories used to model modular exponentiation (up to AC) have the FVP;
- homomorphic encryption $\{\langle x, y\rangle\}_z = \langle \{x\}_z, \{y\}_z\rangle$ does **not** satisfy the FVP.

Finite Variant Property (FVP) [Comon & D., 05]

An equational theory E (represented by a rewrite sytem $\downarrow$ - possibly modulo $\mathsf{E}' = \mathsf{AC}$) has the finite variant property if for any term t, we can compute a finite set of instances $t\sigma_1, \ldots, t\sigma_n$ such that:

$$\{t\sigma\downarrow \mid \sigma \in \Sigma\} = \bigcup_{i=1}^{n} \{t\sigma_i\downarrow\theta \mid \theta \in \Sigma\}$$

where Σ is the set of normalized substitutions.

$\longrightarrow$ variant$(t) = \{t\sigma_1, \ldots, t\sigma_n\}$.

A link with rewriting theory!

Actually, the finite variant property is implied by the termination of **basic narrowing** when $\mathsf{E}' = \emptyset$. A bit more tricky when $\mathsf{E}' = \mathsf{AC}$.

Some existing tools exploiting the FVP

TAMARIN Tamarin prover interactive mode	**Maude-NPA**	DEEPSEC
[Meier et al., 13]	[Escobar et al., 07]	[Cheval et al., 18]

Some existing tools exploiting the FVP

TAMARIN
Tamarin prover interactive mode

Maude-NPA

DEEPSEC

[Meier et al., 13] [Escobar et al., 07] [Cheval et al., 18]

From its input:

```
rule B: [!Key(k), In(x)] --> [Out(senc{h(sdec(x,k))}k)]
```

Tamarin computes:

```
rule (modulo AC) B:
   [ !Key( k ), In( x ) ] --> [ Out( senc(h(z), k) ) ]
  variants (modulo AC)
  1. k = k.4
     x = x.4
     z = sdec(x.4, k.4)

  2. k = x.4
     x = senc(x.5, x.4)
     z = x.5
```

Conclusion

Many UNIF topics are of interest for protocol verification:

- Equational unification and unification modulo theories
- Narrowing
- Higher-Order Unification
- Contraint Solving
- Disunification
- ...

Conclusion

Many UNIF topics are of interest for protocol verification:

- Equational unification and unification modulo theories
- Narrowing
- Higher-Order Unification
- Contraint Solving
- Disunification
- ...

Challenging theory: A useful equational theory on which existing tools behave badly is homomophic encryption (e-voting protocols):

$$\{x\}_{\mathsf{pk}(s)} \star \{y\}_{\mathsf{pk}(s)} = \{x + y\}_{\mathsf{pk}(s)}$$

Conclusion

Many UNIF topics are of interest for protocol verification:

- Equational unification and unification modulo theories
- Narrowing
- Higher-Order Unification
- Contraint Solving
- Disunification
- ...

Challenging theory: A useful equational theory on which existing tools behave badly is homomophic encryption (e-voting protocols):

$$\{x\}_{\mathsf{pk}(s)} \star \{y\}_{\mathsf{pk}(s)} = \{x + y\}_{\mathsf{pk}(s)}$$

Thanks you for listening