A Probablistic Logic for Concrete Security

David Baelde[1], Caroline Fontaine[2], Adrien Koutsos[3], Guillaume Scerri[2], Théo Vignon[2]

[1]Université de Rennes, CNRS, IRISA
[2]Université Paris-Saclay, CNRS, ENS Paris-Saclay, LMF
[3]Inria Paris

April 4, 2024

Protocol Verification

- Protocols (distributed programs)
  - Messaging
  - Login
- Various security properties
  - Confidentiality
  - Privacy
  - Authentication

Critical properties $\Rightarrow$ **How to formally state and prove such properties**?

Running Example

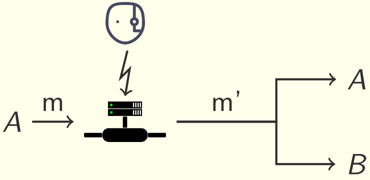
Protocol — Private Authentication

- $A \rightarrow B : \{\mathrm{pk}_A, n_A\}_{\mathrm{pk}_B}$
- $B \rightarrow A : \begin{cases} \{n_B, n_A\}_{\mathrm{pk}_A} & \text{if B receive a valid message} \\ \{n_B, 0^\eta\}_{\mathrm{pk}_A} & \text{otherwise} \end{cases}$

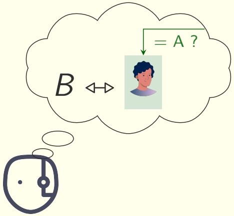
Arbitrary number of interactions.

Attacker Model

Attacker controls the network: can see and alter messages



Goal



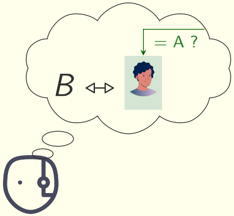
The Computational Model

Model

- messages as bitstrings
- attacker as a Polynomial-time Probabilistic Turing Machine
- security property is a game (we give one of two scenarios to the adversary)

Running Example — Security Property

Our goal is to prove:



Can be modeled in the computational model as:

Idealization of the protocol

$$\phi_N = B^N \mid\mid \underbrace{A \mid\mid \ldots \mid\mid A}_{N \text{ times}} \approx B^N \mid\mid A_1 \mid\mid \ldots \mid\mid A_N$$

Real version of the protocol

The Computational Model

Model

- messages as bitstrings
- attacker as a Polynomial-time Probabilistic Turing Machine
- security property is a game (we give one of two scenarios to the adversary)

Two Flavors of Security

- **concrete security**: for all attackers ☺, Pr(☺ break ϕ_N) $\leq \varepsilon_N$
- **asymptotic security**: for all attackers ☺, Pr(☺ break $\phi_{P(\eta)}$) is negligible in η
  - η: security parameter (e.g. length of keys)
  - negligible: asymptotically small
- concrete security $+\ \varepsilon_\eta$ negligible in $\eta \Rightarrow$ asymptotic security

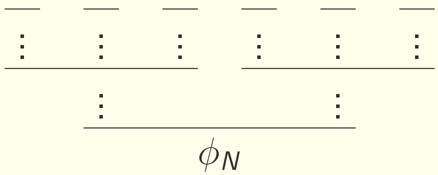
CCSA in a Nutshell

A formal framework for computational security

- indistinguishability predicate: $\sim$
- messages: symbolic terms
  Semantics: bitstrings
- protocol: recursive terms (representing all possible execution of the protocol)
- sound proof system for $\sim$ with support for:
  - generic mathematical reasoning
  - cryptographic reductions, e.g. IND-CCA1, PRF

Definition: Running Example — Privacy of PA

$$\phi_N = \mathsf{frame}_{\mathcal{P}}(N) \sim \mathsf{frame}_{\mathcal{I}}(N)$$



Running Example - Private Authentication in CCSA

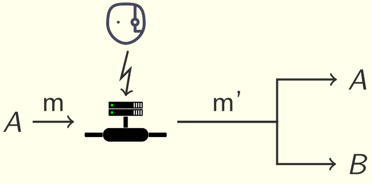
High-Level

Statement of security property:

Definition: Running Example — Privacy of PA

$$\phi_N = \mathsf{frame}_{\mathcal{P}}(N) \sim \mathsf{frame}_{\mathcal{I}}(N)$$

Attacker Model



Execution of the protocol modeled by (mutually) recursive function:

- output(X, N): the output of the agent X
- input(N): the input given by the adversary
- choose(N): choice of the adversary of which agent do something
- frame(N): the knowledge of the adversary

} at step N

Private Authentication in CCSA

The actual terms

attacker computation

$$\mathsf{input}(N) \overset{\mathsf{def}}{=} \underline{att}_i(\mathsf{frame}(N-1)) \qquad \mathsf{choose}(N) \overset{\mathsf{def}}{=} \underline{att}_c(\mathsf{frame}(N-1))$$

$$\mathsf{frame}(N) \overset{\mathsf{def}}{=} \begin{cases} \mathsf{frame}(N-1), \mathsf{output}(\mathsf{choose}(N), N) & \text{if } N \geq 0 \\ \mathtt{pk}_A, \mathtt{pk}_B & \text{if } N = 0 \end{cases}$$

Private Authentication in CCSA

The actual terms

attacker computation

$$\mathsf{input}(N) \overset{\text{def}}{=} \underline{att}_i\,(\mathsf{frame}(N-1)) \qquad \mathsf{choose}(N) \overset{\text{def}}{=} \underline{att}_c\,(\mathsf{frame}(N-1))$$

$$\mathsf{frame}(N) \overset{\text{def}}{=} \begin{cases} \mathsf{frame}(N-1), \mathsf{output}(\mathsf{choose}(N), N) & \text{if } N \geq 0 \\ \mathrm{pk}_A, \mathrm{pk}_B & \text{if } N = 0 \end{cases}$$

Protocol Specific Definitions

$$\mathsf{output}(X, N)$$

For example,

$$\mathsf{output}_{\mathcal{P}}(A, N) \overset{\text{def}}{=} \{\mathrm{pk}_A, n_A\ N\}_{\mathrm{pk}_B}$$

$$\mathsf{output}_{\mathcal{I}}(A, N) \overset{\text{def}}{=} \{\mathrm{pk}'_A\ N, n_A\ N\}_{\mathrm{pk}_B}$$

Limit of the CCSA Approach

On the running example

Best we can do

$$\tilde{\forall}(N : \mathbb{N}), \text{const}(N) \quad \overset{\sim}{\Rightarrow} \quad \text{frame}_{\mathcal{P}}(N) \sim \text{frame}_{\mathcal{I}}(N)$$

N is independent form η

Meaning

interpretation of terms as bitstrings

$$\forall N, \underbrace{\left|\Pr(\text{☺}(\llbracket \text{frame}_{\mathcal{P}} \rrbracket^{\eta}(N)) = 1) - \Pr(\text{☺}(\llbracket \text{frame}_{\mathcal{I}} \rrbracket^{\eta}(N)) = 1)\right|}_{\text{denoted } \mathbf{Adv}_{\text{☺}}(\text{frame}_{\mathcal{P}}(N) \sim \text{frame}_{\mathcal{I}}(N))} \text{ is negligible in } \eta$$

More precisely,

$$\forall N, \exists f \text{ negligible such as for all } \eta, \mathbf{Adv}_{\text{☺}}(\text{frame}_{\mathcal{P}}(N) \sim \text{frame}_{\mathcal{I}}(N)) \leq f(\eta)$$

Limit of the CCSA Approach

On the running example

Best we can do

$$\tilde{\forall}(N : \mathbb{N}),\ \mathrm{const}(N) \quad \tilde{\Rightarrow} \quad \mathrm{frame}_{\mathcal{P}}(N) \sim \mathrm{frame}_{\mathcal{I}}(N)$$

N is independent form η

Meaning

- Parametric Security (CCSA):

$$\forall N, \exists f \text{ negligible such as for all } \eta,\ \mathbf{Adv}_{\text{☺}}(\mathrm{frame}_{\mathcal{P}}(N) \sim \mathrm{frame}_{\mathcal{I}}(N)) \leq f(\eta)$$

- Asymptotic Security:

$$\exists f \text{ negligible such that for all } \eta, N = P(\eta),\ \mathbf{Adv}_{\text{☺}}(\mathrm{frame}_{\mathcal{P}}(N) \sim \mathrm{frame}_{\mathcal{I}}(N)) \leq f(\eta)$$

Limitations of the CCSA Approach

- it only proves asymptotic results ⇒ not concrete security
- due to const(N), N cannot depend polynomially on η ⇒ not the typical asymptotic security

Limit of the CCSA Approach

On the running example

Best we can do

$$\tilde{\forall}(N : \mathbb{N}), \text{const}(N) \quad \overset{\sim}{\Rightarrow} \quad \underbrace{\text{frame}_{\mathcal{P}}(N) \sim \text{frame}_{\mathcal{I}}(N)}_{\phi_N}$$

$\Rightarrow$ Where does const(N) come form?

Induction

$$\begin{array}{l}\textsc{Induction}\\ \dfrac{\vdash \phi_0 \qquad \vdash \tilde{\forall}(N : \mathbb{N}), \text{const}(N) \quad \overset{\sim}{\Rightarrow} \quad \phi_N \quad \overset{\sim}{\Rightarrow} \quad \phi_{N+1}}{\vdash \tilde{\forall}(N : \mathbb{N}), \text{const}(N) \quad \overset{\sim}{\Rightarrow} \quad \phi_N}\end{array}$$

To avoid this:

$$\begin{array}{lccccccc} & u_1 & \sim & u_2 & \sim & \dots & \sim & u_{N+1}\\ & & \uparrow & & \uparrow & & \uparrow & \\ \textbf{Adv}: & & f_1 & & f_2 & & f_N & \end{array}$$

Total Adv : $g_N = \sum_{i=1}^{N} f_i$

Problem: is g_η negligible in η?

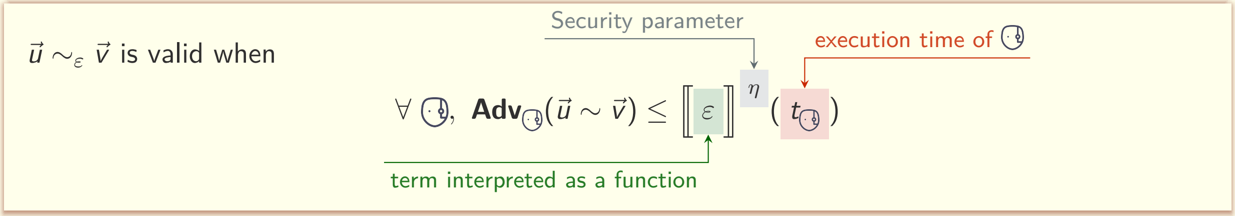
$\Rightarrow$ not necessarily

$\Rightarrow$ need uniform bound for the f

Contribution: Switch to Concrete Security

Predicate

- Adaptation of $\sim$:



$\vec{u} \sim_{\varepsilon} \vec{v}$ is valid when

$$\forall \text{☺},\ \mathbf{Adv}_{\text{☺}}(\vec{u} \sim \vec{v}) \leq \underbrace{[\![\varepsilon]\!]}_{\text{term interpreted as a function}}^{\overset{\text{Security parameter}}{\eta}} (\overset{\text{execution time of ☺}}{t_{\text{☺}}})$$

- this required us to come up with a precise Turing machine cost model
- adaption of the proof system for $\sim_{\varepsilon}$

Switch to Concrete Security

Rules

adversary can compute f in time t_f

$$\textsc{FA}\ \frac{\vdash x \sim_{\varepsilon} y \qquad \vdash \boxed{\mathsf{adv}_{t_f}(f)}}{\vdash f(x) \sim_{\lambda t.\varepsilon(t+t_f)} f(y)}$$

$$\textsc{Case-Study}\ \frac{\vdash b_l, v_l \sim_{\varepsilon_1} b_r, v_r \qquad \vdash b_l, w_l \sim_{\varepsilon_2} b_r, w_r}{\vdash \begin{array}{c} b_l \\ / \quad \backslash \\ v_l \qquad w_l \end{array} \sim_{\varepsilon_1+\varepsilon_2} \begin{array}{c} b_r \\ / \quad \backslash \\ v_r \qquad w_r \end{array}}$$

Switch to Concrete Security

Induction rule

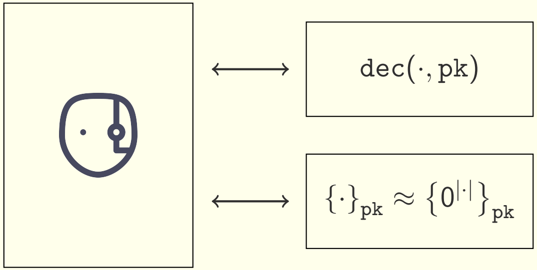
$$\textsc{Induction}\;\dfrac{\vdash \overset{\phi_0}{\boxed{u(0) \sim_{\varepsilon(0)} v(0)}} \qquad \vdash \tilde{\forall}(N : \mathbb{N}), \overset{\phi_N}{\boxed{u(N) \sim_{\varepsilon(N)} v(N)}} \quad \tilde{\Rightarrow} \quad \overset{\phi_{N+1}}{\boxed{u(N+1) \sim_{\varepsilon(N+1)} v(N+1)}}}{\vdash \tilde{\forall}(N : \mathbb{N}), u(N) \sim_{\underset{\text{Explicit dependency on } N}{\boxed{\varepsilon(N)}}} v(N)}$$

$\Rightarrow$ This is the induction that we wanted

Switch to Concrete Security

Cryptographic rules

Cryptographic Game



$\mathtt{dec}(\cdot, \mathtt{pk})$

$\{\cdot\}_{\mathrm{pk}} \approx \{0^{|\cdot|}\}_{\mathrm{pk}}$

CCSA Rule

decryption oracle "well-used"

$$\frac{\vdash \mathsf{adv}_{t_m}(m,b) \qquad \vdash [\phi^{\mathrm{pk}}_{\mathsf{dec}}(m,b)]}{\vdash \begin{array}{c} b \\ / \quad \backslash \\ \{m\}_{\mathrm{pk}} \qquad w \end{array} \sim_{\lambda t.\ \varepsilon_{\mathsf{CCA1}}(t + t_m)} \begin{array}{c} b \\ / \quad \backslash \\ \left\{0^{|m|}\right\}_{\mathrm{pk}} \qquad w \end{array}}\ \mathrm{CCA1}$$

cryptographic upper-bound

reduction time-shift

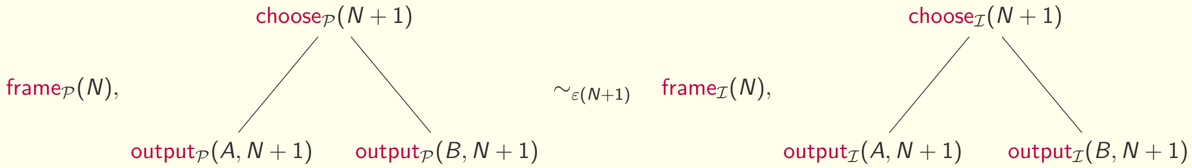
Proof of Private Authentication in Concrete CCSA

Induction

We want to prove $\tilde{\forall} N, \mathsf{frame}_{\mathcal{P}}(N) \sim_{\varepsilon(N)} \mathsf{frame}_{\mathcal{I}}(N)$ for some ε
By induction, assume that:

$$\mathsf{frame}_{\mathcal{P}}(N) \sim_{\varepsilon(N)} \mathsf{frame}_{\mathcal{I}}(N)$$

We must prove:



$\mathsf{frame}_{\mathcal{P}}(N)$, $\mathsf{choose}_{\mathcal{P}}(N+1)$ $\mathsf{output}_{\mathcal{P}}(A, N+1)$ $\mathsf{output}_{\mathcal{P}}(B, N+1)$ $\sim_{\varepsilon(N+1)}$ $\mathsf{frame}_{\mathcal{I}}(N)$, $\mathsf{choose}_{\mathcal{I}}(N+1)$ $\mathsf{output}_{\mathcal{I}}(A, N+1)$ $\mathsf{output}_{\mathcal{I}}(B, N+1)$

⇒ application of the case-study rule

Proof of Private Authentication in Concrete CCSA

Shape of idiomatic proofs

Proof looks like this:

$$\dfrac{\begin{array}{cc} \dfrac{\vdash \mathsf{frame}_{\mathcal{P}}(N) \ \sim_{\varepsilon(N)} \mathsf{frame}_{\mathcal{I}}(N)}{\vdots} & \qquad \dfrac{\vdash \mathsf{frame}_{\mathcal{P}}(N) \ \sim_{\varepsilon(N)} \mathsf{frame}_{\mathcal{I}}(N)}{\vdots} \end{array}}{\vdash \mathsf{frame}_{\mathcal{P}}(N+1) \ \sim_{\varepsilon(N+1)} \mathsf{frame}_{\mathcal{I}}(N+1)} \ \textsc{Case-Study}$$

Still, not good enough:

$$\text{final bound: } \varepsilon(N+1) = 2 \times \varepsilon(N) + \varepsilon_{aux}$$

⇒ exponential in N, thus not expected asymptotic security

Proof of Private Authentication in Concrete CCSA

Wanted proof-shape

We want a proof like this:

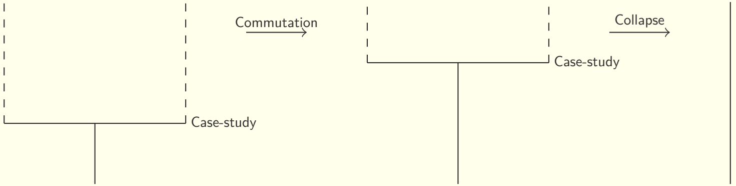
$$\cfrac{\cfrac{\vdash \mathsf{frame}_{\mathcal{P}}(N) \ \sim_{\varepsilon(N)} \mathsf{frame}_{\mathcal{I}}(N)}{\vdots}}{\vdash \mathsf{frame}_{\mathcal{P}}(N+1) \ \sim_{\varepsilon(N+1)} \mathsf{frame}_{\mathcal{I}}(N+1)}$$

Final bound: $\varepsilon(N+1) = \varepsilon(N) + \varepsilon_{aux}$

Yields the expected asymptotic security.
It is possible?
$\Rightarrow$ Yes, thanks to the generalization of the rules
Problem: Those proofs a bit less natural.

Proof Transformation

Overview



- 3 kind of rules:
  - "ascending" rules: Case-study, ...
  - "descending" rules: most of the other relevant rules
  - "leaf" rules: rules without premises
- transformation $\blacktriangleright_{\mathsf{AD}}$ (Commutation): commuting ascending/descending rules
- transformation $\blacktriangleright_{\mathsf{col}}$ (Collapse) $\Rightarrow$ remove the case-studies

Proof Transformation

Shape

Ideal $\blacktriangleright_{\mathsf{AD}}$ Rule

$$\dfrac{\dfrac{\Pi_1 \quad \mathsf{Aux}_R}{\cdot}\,\textsc{Descending} \qquad \Pi_2}{\cdot}\,\text{CS} \quad \blacktriangleright_{\mathsf{AD}} \quad \dfrac{\dfrac{\Pi_1 \quad \Pi_2}{\cdot}\,\text{CS} \qquad \mathsf{Aux}_R}{\cdot}\,\textsc{Descending}$$

Proof Transformation

Shape

Ic

Actual $\blacktriangleright_{\mathsf{AD}}$ Rule

$$\frac{\dfrac{\Pi_1 \quad \mathsf{Aux}_R}{\cdot}\,\textsc{Descending} \qquad \Pi_2}{\cdot}\,\mathrm{CS} \quad \blacktriangleright_{\mathsf{AD}} \quad \frac{\dfrac{\dfrac{\dfrac{\dfrac{\Pi_1}{\cdot}\,\mathrm{B}^* \quad \dfrac{\Pi_2}{\cdot}\,\mathrm{B}^*}{\cdot}\,\mathrm{CS}}{\cdot}\,\mathrm{B}^* \qquad \dfrac{\mathsf{Aux}_R}{\cdot}\,\mathrm{B}^*}{\cdot}\,\textsc{Descending}}{\cdot}\,\mathrm{B}^*$$

Proof Transformation

Shape

$\blacktriangleright_{\text{col}}$ Rule

$$\dfrac{\dfrac{\overset{\text{IH}}{\vdots} \qquad \text{Aux}_0}{\vdots}\;\text{B} \qquad \dfrac{\overset{\text{IH}}{\vdots} \qquad \text{Aux}_1}{\vdots}\;\text{B}}{\vdots}\;\text{CS} \quad \blacktriangleright_{\text{col}} \quad \dfrac{\dfrac{\overset{\text{IH}}{\vdots} \qquad \overset{\dfrac{\text{Aux}_0 \ \ \text{Aux}_1}{}\;A^*}{\vdots}}{\vdots}\;\text{B} \qquad \overset{A^*}{\vdots}}{\vdots}\;\text{R}$$

Proof Transformation

Transformation result

Proposition: Transformation to a proof of asymptotic security

All nice proofs can be transformed into a proof that yield asymptotic security

some sub-class of proof

Corolary: Asymptotic Security

Nice proofs yield asymptotic security

Conclusion

Contributions:

- CCSA with concrete bounds,
- with the possibility to explicitly extract security bounds,
- with asymptotic security for polynomial number of sessions,
- with (limited) support for the previous way of writing proofs.

What's next:

- add this to Squirrel
- we have explicit probabilities: extend CCSA to non-negligible probabilities (e.g. PAKE).
- we have concrete security: use CCSA to prove primitives (e.g. KEM-DEM).

Conclusion

Contributions:

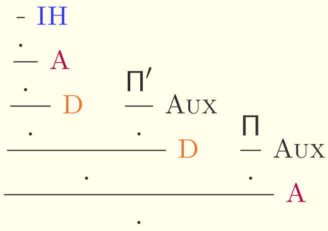
- CCSA with concrete bounds,
- with the possibility to explicitly extract security bounds,
- with asymptotic security for polynomial number of sessions,
- with (limited) support for the previous way of writing proofs.

What's next:

- add this to Squirrel
- we have explicit probabilities: extend CCSA to non-negligible probabilities (e.g. PAKE).
- we have concrete security: use CCSA to prove primitives (e.g. KEM-DEM).

Thank you for your attention

A Nice Class of Proofs



- Only ascending / descending under a induction hypothesis
- Auxiliary proofs (Π, Π') have nice ($\simeq$ polynomial) constraints ($=$ advantage/time bounds)