Vote-Independence: A Powerful Privacy Notion for Voting Protocols

Jannik Dreier, Pascal Lafourcade, Yassine Lakhnech

Université Grenoble 1, CNRS, Verimag

FPS 2011: May 13, 2011

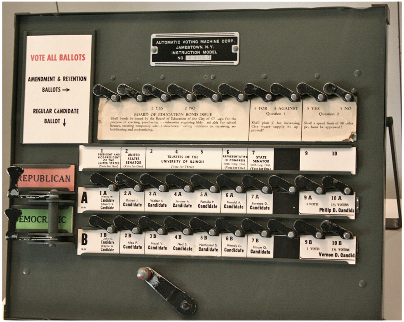
Plan

1. Introduction
   - What is electronic voting?
   - An Attack on Privacy in Helios
2. Intuitive Definitions
   - Privacy
   - Vote-Independence
3. Formal Definitions
4. Analysis and Case Studies
5. Conclusion

Plan

Plan

Voting machines are not a recent technology



They have been in use in the US for over 100 years!

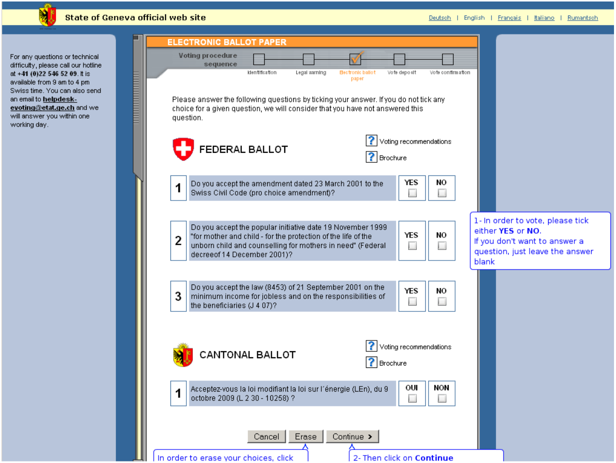
Electronic voting machines. . .

. . . are used all over the world

Internet voting

Available in

- Estonia
- France
- Switzerland
- ...



Security Requirements

Fairness

Individual Verifiability

Eligibility

Universal Verifiability

Security Requirements

Privacy

Robustness

Receipt-Freeness

Coercion-Resistance

Security Requirements

Fairness

Individual Verifiability

Eligibility

Universal Verifiability

Security Requirements

Privacy

Robustness

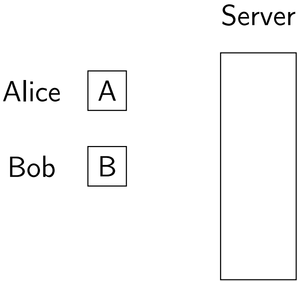
Receipt-Freeness

Coercion-Resistance

Plan

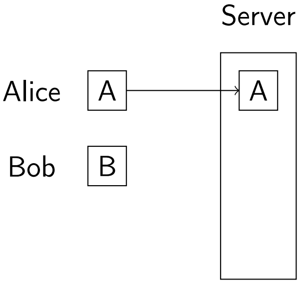
Attack on Privacy in Helios [?]

Helios [?] is a web based open-source voting system based on homomorphic encryption.



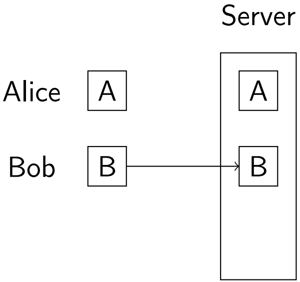
Attack on Privacy in Helios [?]

Helios [?] is a web based open-source voting system based on homomorphic encryption.



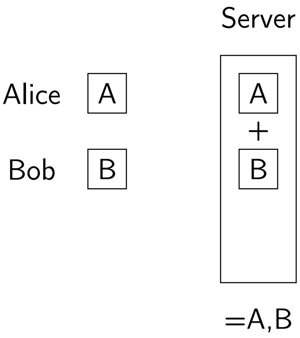
Attack on Privacy in Helios [?]

Helios [?] is a web based open-source voting system based on homomorphic encryption.



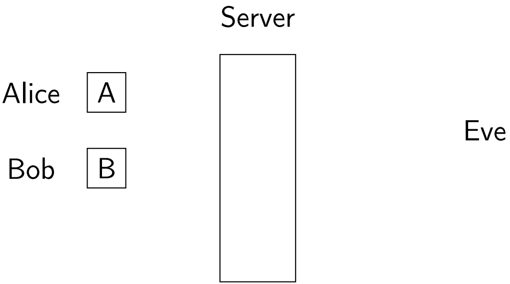
Attack on Privacy in Helios [?]

Helios [?] is a web based open-source voting system based on homomorphic encryption.



Attack on Privacy in Helios [?]

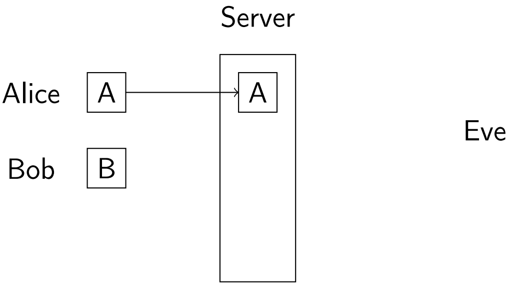
Eve can attack Alice's privacy by copying her vote:



To prevent this attack, we have to enforce *Vote-Independence*.

Attack on Privacy in Helios [?]

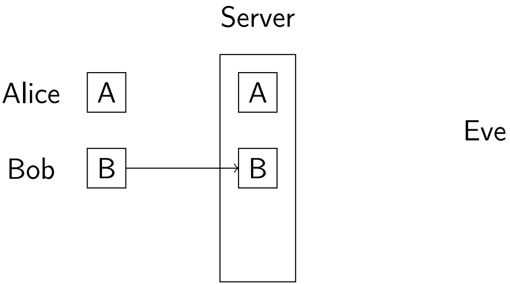
Eve can attack Alice's privacy by copying her vote:



To prevent this attack, we have to enforce *Vote-Independence*.

Attack on Privacy in Helios [?]

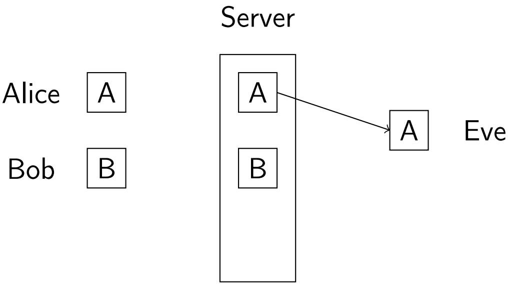
Eve can attack Alice's privacy by copying her vote:



To prevent this attack, we have to enforce *Vote-Independence*.

Attack on Privacy in Helios [?]

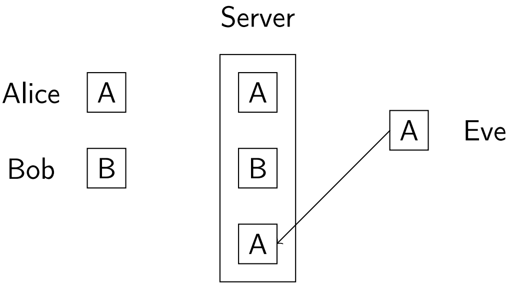
Eve can attack Alice's privacy by copying her vote:



To prevent this attack, we have to enforce *Vote-Independence*.

Attack on Privacy in Helios [?]

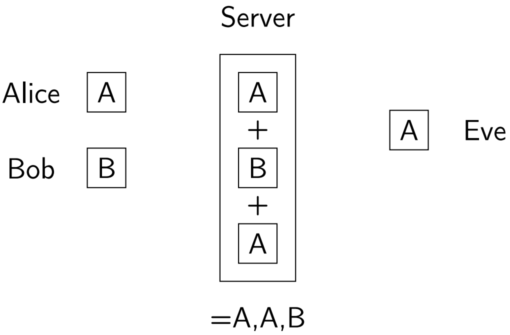
Eve can attack Alice's privacy by copying her vote:



To prevent this attack, we have to enforce *Vote-Independence*.

Attack on Privacy in Helios [?]

Eve can attack Alice's privacy by copying her vote:



To prevent this attack, we have to enforce *Vote-Independence*.

Plan

Plan

Defining Vote-Privacy [?]

Main idea: Observational equivalence between two situations.

	Alice	Bob
Vote	A	B

$$\approx_l$$

Vote	B	A

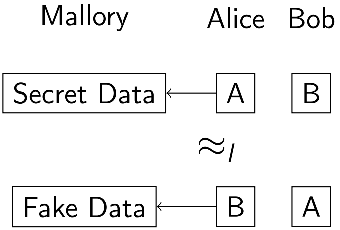
Defining Receipt-Freeness [?]

Again: Observational equivalence between two situations, but Alice tries to create a receipt or a fake.

Mallory	Alice	Bob
	A	B
	$\approx_l$	
	B	A

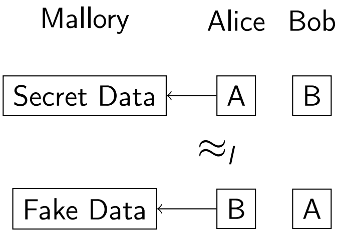
Defining Receipt-Freeness [?]

Again: Observational equivalence between two situations, but Alice tries to create a receipt or a fake.



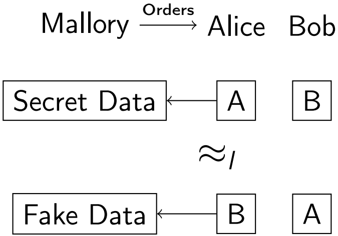
Defining Coercion-Resistance [?]

Observational equivalence between two situations, but Alice is under control by Mallory or only pretends to be so.



Defining Coercion-Resistance [?]

Observational equivalence between two situations, but Alice is under control by Mallory or only pretends to be so.



Plan

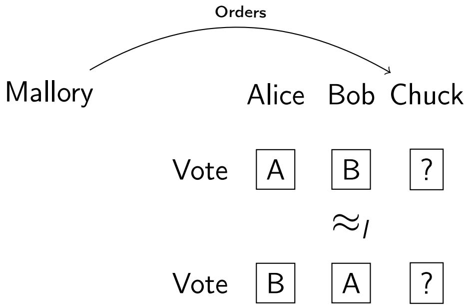
Defining Vote-Independence

Main idea: Privacy, but with a voter under control of the attacker. If he can relate his vote to e.g. Alice's vote, Mallory can distinguish both sides.

Mallory		Alice	Bob
	Vote	A	B
			$\approx_l$
	Vote	B	A

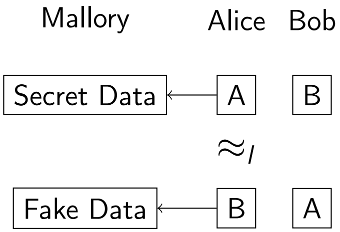
Defining Vote-Independence

Main idea: Privacy, but with a voter under control of the attacker. If he can relate his vote to e.g. Alice's vote, Mallory can distinguish both sides.



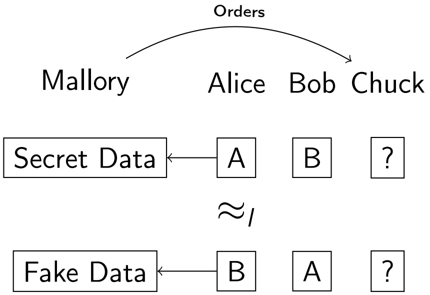
Vote-Independence with Passive Collaboration

"Receipt-Freeness with Chuck":



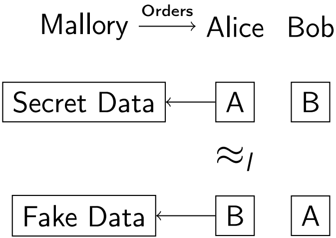
Vote-Independence with Passive Collaboration

"Receipt-Freeness with Chuck":



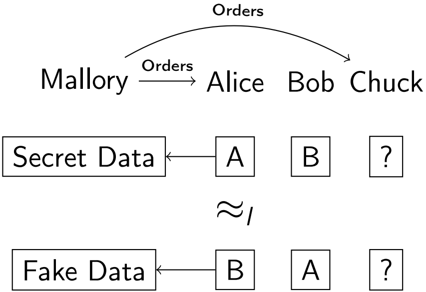
Vote-Independence with active Collaboration

"Coercion-Resistance with Chuck":



Vote-Independence with active Collaboration

"Coercion-Resistance with Chuck":



Plan

The Applied Pi Calculus [?]

Syntax

$P, Q, R :=$	processes
0	null process
$P \mid Q$	parallel composition
$!P$	replication
$\nu n.P$	name restriction ("new")
$\mathtt{if}\ M = N\ \mathtt{then}\ P\ \mathtt{else}\ Q$	conditional
$\mathtt{in}(u, x)$	message input
$\mathtt{out}(u, x)$	message output
$\{^M/_x\}$	substitution

Modeling a voting protocol

Definition (Voting Process [?])

A voting process is a closed plain process

$$VP \equiv \nu \tilde{n}.(V\sigma_1 | \ldots | V\sigma_n | A_1 | \ldots | A_m).$$

We define an evaluation context S which is like VP, but has a hole instead of three $V\sigma_i$, and an evaluation context S' which is like VP, but has a hole instead of two $V\sigma_i$.

Vote-Privacy: The formal definition

Definition (Vote-Privacy [?])

A voting process respects *Vote-Privacy (P)* if for all votes a and b we have

$$S'\left[V_A\left\{{}^{a}/_{v}\right\}\middle|V_B\left\{{}^{b}/_{v}\right\}\right] \approx_l S'\left[V_A\left\{{}^{b}/_{v}\right\}\middle|V_B\left\{{}^{a}/_{v}\right\}\right]$$

Vote-Independence (without Collaboration): The formal definition

Definition (Vote-Independence)

A voting process respects *Vote-Independence (VI)* if for all votes a and b we have

$$S\left[V_A\left\{{}^{a}/_{v}\right\}|V_B\left\{{}^{b}/_{v}\right\}|V_C^{c_1,c_2}\right] \approx_l S\left[V_A\left\{{}^{b}/_{v}\right\}|V_B\left\{{}^{a}/_{v}\right\}|V_C^{c_1,c_2}\right]$$

Receipt-Freeness: The formal definition

Definition (Receipt-Freeness [?])

A voting process respects *Receipt-Freeness (RF)* if there exists a closed plain process V' such that for all votes a and c we have

$$V'^{\backslash out(chc,\cdot)} \approx_l V_A \{^a/_v\}$$

and

$$S'\left[V_A \{^b/_v\}^{chc} | V_B \{^a/_v\}\right] \approx_l S'\left[V' | V_B \{^b/_v\}\right]$$

Vote-Independence with Passive Collaboration: The formal definition

Definition (Vote-Independence with Passive Collaboration)

A voting process respects *Vote-Independence with Passive Collaboration (VI-PC)* if there exists a closed plain process V' such that for all votes a and c we have

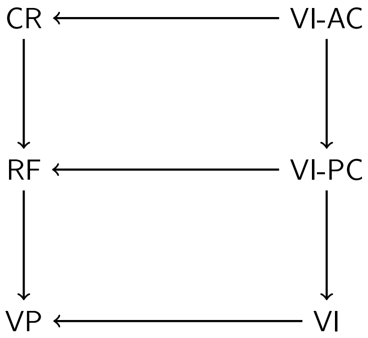
$$V'^{\backslash out(chc,\cdot)} \approx_l V_A\{^a/_v\}$$

and

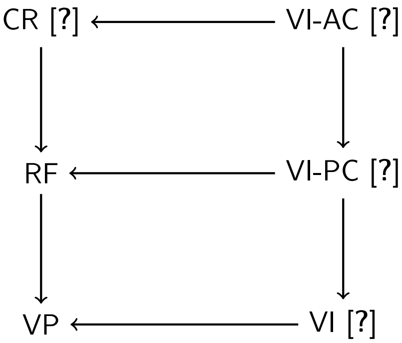
$$S\left[V_A\{^b/_v\}^{chc}|V_B\{^a/_v\}|V_C^{c_1,c_2}\right] \approx_l S\left[V'|V_B\{^b/_v\}|V_C^{c_1,c_2}\right]$$

Plan

Relations among the notions



Examples



Plan

Conclusion

- Attack on Helios
- Extended threat model
- Formal definition of "Vote-Independence"
- Strictly stronger than standard Vote-Privacy
- Generalized to passive and active collaboration
- Case studies: even Coercion-Resistant protocols may not ensure Vote-Independence

Future Work

- Generalized definition of voting protocols
- Tools to automate and/or verify the proofs (at least partly)
- Computational definition

Thank you for your attention!

Questions?

Ben Adida, Olivier De Marneffe, Olivier Pereira, and Jean-Jacques Quisquater.
Electing a university president using open-audit voting: analysis of real-world use of helios.
In *Proceedings of the 2009 conference on Electronic voting technology/workshop on trustworthy elections*, EVT/WOTE'09, pages 10–10, Berkeley, CA, USA, 2009. USENIX Association.

Martín Abadi and Cédric Fournet.
Mobile values, new names, and secure communication.
In *Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '01, pages 104–115, New York, 2001. ACM.

Jens-Matthias Bohli, Jörn Müller-Quade, and Stefan Röhrich.

Bingo voting: Secure and coercion-free voting using a trusted random number generator.
In Ammar Alkassar and Melanie Volkamer, editors, *E-Voting and Identity*, volume 4896 of *Lecture Notes in Computer Science*, pages 111–124. Springer Berlin / Heidelberg, 2007.

Stéphanie Delaune, Steve Kremer, and Mark Ryan.
Verifying privacy-type properties of electronic voting protocols.
Journal of Computer Security, 17:435–487, December 2009.

Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta.
A practical secret voting scheme for large scale elections.
In Jennifer Seberry and Yuliang Zheng, editors, *Advances in Cryptology – AUSCRYPT '92*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer Berlin / Heidelberg, 1992.

Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang, and Seungjae Yoo.
Providing receipt-freeness in mixnet-based voting protocols.
In Jong In Lim and Dong Hoon Lee, editors, *Information Security and Cryptology - ICISC 2003*, volume 2971 of *Lecture Notes in Computer Science*, pages 245–258. Springer Berlin / Heidelberg, 2004.

Tatsuaki Okamoto.
An electronic voting scheme.
In *Proceedings of the IFIP World Conference on IT Tools*, pages 21–30, 1996.

Ben Smyth and Veronique Cortier.
Attacking and fixing helios: An analysis of ballot secrecy.
Cryptology ePrint Archive, Report 2010/625, 2010.
http://eprint.iacr.org/.

Coercion-Resistance: The formal definition

Definition (Coercion-Resistance [?])

A voting process respects *Coercion-Resistance (CR)* if there exists a closed plain process V' such that for any $C = \nu c_1.\nu c_2.(\_|P)$ satisfying $\tilde{n} \cap fn(C) = \emptyset$ and $S'\left[C\left[V_A\left\{{}^{?}/_{v}\right\}^{c_1,c_2}\right]|V_B\left\{{}^{a}/_{v}\right\}\right] \approx_l S'\left[V_A\left\{{}^{b}/_{v}\right\}^{chc}|V_B\left\{{}^{a}/_{v}\right\}\right]$ and for all votes a and c we have

$$C\left[V'\right]^{\backslash out(chc,\cdot)} \approx_l V_A\left\{{}^{a}/_{v}\right\}$$

and

$$S'\left[C\left[V_A\left\{{}^{?}/_{v}\right\}^{c_1,c_2}\right]|V_B\left\{{}^{a}/_{v}\right\}\right] \approx_l S'\left[C\left[V'\right]|V_B\left\{{}^{b}/_{v}\right\}\right]$$

Vote-Independence with Active Collaboration: The formal definition

Definition (Vote-Independence with Active Collaboration)

A voting process respects *Vote-Independence with Active Collaboration (VI-AC)* if there exists a closed plain process V' such that for any $C = \nu c_1.\nu c_2.(\_|P)$ satisfying $\tilde{n} \cap fn(C) = \emptyset$ and

$$S\left[C\left[V_A\left\{?/v\right\}^{c_1,c_2}\right]|V_B\left\{a/v\right\}|V_C^{c_3,c_4}\right] \approx_l S\left[V_A\left\{b/v\right\}^{chc}|V_B\left\{a/v\right\}|V_C^{c_3,c_4}\right]$$

and for all votes a and c we have

- $C\left[V'\right]^{\backslash out(chc,\cdot)} \approx_l V_A\left\{a/v\right\}$
- $S\left[C\left[V_A\left\{?/v\right\}^{c_1,c_2}\right]|V_B\left\{a/v\right\}|V_C^{c_3,c_4}\right] \approx_l S\left[C\left[V'\right]|V_B\left\{b/v\right\}|V_C^{c_3,c_4}\right]$

Definition (Process P^{ch} [?])

Let P be a process and ch be a channel. We define P^{ch} as follows:

- $0^{ch} \triangleq 0$,
- $(P|Q)^{ch} \triangleq P^{ch}|Q^{ch}$,
- $(\nu n.P)^{ch} \triangleq \nu n.\mathrm{out}(ch, n).P^{ch}$ when n is a name of base type,
- $(\nu n.P)^{ch} \triangleq \nu n.P^{ch}$ otherwise,
- $(\mathtt{in}(u, x).P)^{ch} \triangleq \mathtt{in}(u, x).\mathtt{out}(ch, x).P^{ch}$ when x is a variable of base type,
- $(\mathtt{in}(u, x).P)^{ch} \triangleq \mathtt{in}(u, x).P^{ch}$ otherwise,
- $(\mathtt{out}(u, M).P)^{ch} \triangleq \mathtt{out}(u, M).P^{ch}$,
- $(!P)^{ch} \triangleq !P^{ch}$,
- $(\mathtt{if}\ M = N\ \mathtt{then}\ P\ \mathtt{else}\ Q)^{ch} \triangleq \mathtt{if}\ M = N\ \mathtt{then}\ P^{ch}\ \mathtt{else}\ Q^{ch}$.

Definition (Process P^{c_1,c_2} [?])

Let P be a process, c_1, c_2 channels. We define P^{c_1,c_2} as follows:

- $0^{c_1,c_2} \triangleq 0$,
- $(P|Q)^{c_1,c_2} \triangleq P^{c_1,c_2}|Q^{c_1,c_2}$,
- $(\nu n.P)^{c_1,c_2} \triangleq \nu n.\mathtt{out}(c_1, n).P^{c_1,c_2}$ if n is a name of base type,
- $(\nu n.P)^{c_1,c_2} \triangleq \nu n.P^{c_1,c_2}$ otherwise,
- $(\mathtt{in}(u, x).P)^{c_1,c_2} \triangleq \mathtt{in}(u, x).\mathtt{out}(c_1, x).P^{c_1,c_2}$ if x is a variable of base type & x is a fresh variable,
- $(\mathtt{in}(u, x).P)^{c_1,c_2} \triangleq \mathtt{in}(u, x).P^{c_1,c_2}$ otherwise,
- $(\mathtt{out}(u, M).P)^{c_1,c_2} \triangleq \mathtt{in}(c_2, x).\mathtt{out}(u, x).P^{c_1,c_2}$,
- $(!P)^{c_1,c_2} \triangleq !P^{c_1,c_2}$,
- $(\mathtt{if}\ M = N\ \mathtt{then}\ P\ \mathtt{else}\ Q)^{c_1,c_2} \triangleq \mathtt{in}(c_2, x).\mathtt{if}\ x = \mathrm{true}$ $\mathtt{then}\ P^{c_1,c_2}\ \mathtt{else}\ Q^{c_1,c_2}$ where x is a fresh variable and true is

Definition (Process $A^{\setminus out(ch,\cdot)}$ [?])

Let A be an extended process. We define the process $A^{\setminus out(ch,\cdot)}$ as $\nu ch.(A|!\text{in}(ch,x))$.

Definition (Equivalence in a Frame)

Two terms M and N are equal in the frame ϕ, written $(M = N)\phi$, if and only if $\phi \equiv \nu\tilde{n}.\sigma$, $M\sigma = N\sigma$, and $\{\tilde{n}\} \cap (fn(M) \cup fn(N)) = \emptyset$ for some names $\tilde{n}$ and some substitution σ.

Definition (Static Equivalence ($\approx_s$))

Two closed frames ϕ and ψ are statically equivalent, written $\phi \approx_s \psi$, when dom(ϕ) =dom(ψ) and when for all terms M and N $(M = N)\phi$ if and only if $(M = N)\psi$. Two extended processes A and B are statically equivalent ($A \approx_s B$) if their frames are statically equivalent.

Definition (Labelled Bisimilarity ($\approx_l$))

Labelled bisimilarity is the largest symmetric relation $\mathcal{R}$ on closed extended processes, such that $A \mathcal{R} B$ implies

1. $A \approx_s B$,
2. if $A \rightarrow A'$, then $B \rightarrow B'$ and $A' \mathcal{R} B'$ for some B',
3. if $A \xrightarrow{\alpha} A'$ and $fv(\alpha) \subseteq \text{dom}(A)$ and $bn(\alpha) \cap fn(B) = \emptyset$, then $B \rightarrow^* \xrightarrow{\alpha} \rightarrow^* B'$ and $A' \mathcal{R} B'$ for some B'.