Authentication Using Keystroke Timing

Noam Zilberstein and Jonathan Chen

December 2, 2014

Introduction
- Project Motivation
- Project Goals
- Related Work

The Protocol
- The Basic Protocol
- The Zero Knowledge Protocol

Challenges and Limitations

Demo

Project Motivation

- Authenticate users without password

Project Motivation

- Authenticate users without password
- ‘Hardware’ authentication factor without the need of hardware

Project Motivation

- Authenticate users without password
- 'Hardware' authentication factor without the need of hardware
- Login system that is easier and more secure than what is available today

Project Goals

- Develop a prototype login system

Project Goals

- Develop a prototype login system
  - Authentication using keystroke dynamics

Project Goals

- Develop a prototype login system
  - Authentication using keystroke dynamics
  - Ease of use (user only needs to type their name)

Project Goals

- Develop a prototype login system
  - Authentication using keystroke dynamics
  - Ease of use (user only needs to type their name)
- Make the protocol zero-knowledge

Project Goals

- Develop a prototype login system
  - Authentication using keystroke dynamics
  - Ease of use (user only needs to type their name)
- Make the protocol zero-knowledge
  - Keystroke timings are never transmitted; can't be compromised

Project Goals

- Develop a prototype login system
  - Authentication using keystroke dynamics
  - Ease of use (user only needs to type their name)
- Make the protocol zero-knowledge
  - Keystroke timings are never transmitted; can't be compromised
- Determine feasibility of a full-scale implementation

Related Work

- 1997 Monrose and Rubin, authentication scheme using keystroke dynamics

Related Work

- 1997 Monrose and Rubin, authentication scheme using keystroke dynamics
- 2005 Araújo et al, authentication scheme with $\sim 99\%$ accuracy

Related Work

- 1997 Monrose and Rubin, authentication scheme using keystroke dynamics
- 2005 Araújo et al, authentication scheme with $\sim 99\%$ accuracy
- 2010 Stefan and Yao, keystroke dynamic authentication against synthetic forgeries

Related Work

- 1997 Monrose and Rubin, authentication scheme using keystroke dynamics
- 2005 Araújo et al, authentication scheme with $\sim 99\%$ accuracy
- 2010 Stefan and Yao, keystroke dynamic authentication against synthetic forgeries
- But is it possible to have a highly accurate system that is zero-knowledge?

Basic Protocol (Account Creation)

Prover (Client)		**Verifier (Server)**
User types name (A) in web client. Keystroke vector (V) is recorded	$\xrightarrow{(A,V)}$	(A, V) is stored in the database

Note: Very insecure! Everything is transmitted in the clear

Basic Protocol (Authentication)

Prover (Client)		**Verifier (Server)**
User types name (A) in web client. Keystroke vector ($\tilde{V}$) is recorded	$\xrightarrow{(A,\tilde{V})}$	Lookup key A in the database to get V
Report result	$\xleftarrow{f(V,\tilde{V})}$	Send decision

Where $f(V, \tilde{V}) \in \{0, 1\}$ decides if the (noisy) timing vector $\tilde{V}$ is close enough to the expected timing vector V
Note: Also very insecure!

Zero Knowledge Protocol (Adversary)

- Capabilities:

Zero Knowledge Protocol (Adversary)

- Capabilities:
  - Can view exchanges between client and server

Zero Knowledge Protocol (Adversary)

- Capabilities:
  - Can view exchanges between client and server
  - Has access to A (name of the user)

Zero Knowledge Protocol (Adversary)

- Capabilities:
  - Can view exchanges between client and server
  - Has access to A (name of the user)
- Cannot authenticate without the client's keystroke timing V

Zero Knowledge Protocol (Adversary)

- Capabilities:
  - Can view exchanges between client and server
  - Has access to A (name of the user)
- Cannot authenticate without the client's keystroke timing V
- Key assumption: Discrete log is hard

Zero Knowledge Protocol (Account Creation)

Prover (Client)		**Verifier (Server)**
Choose some generator g	$\xrightarrow{g}$	Remember g
User types name (A) in web client. Keystroke vector (V) is recorded. Let $h = g^{f(V)}$	$\xrightarrow{h}$	(A, g, h) is stored in the database

Data is transmitted in the clear. We don't care

Zero Knowledge Protocol (Authentication)

Prover (Client)		**Verifier (Server)**
User types name (A) in web client. Keystroke vector ($\tilde{V}$) is recorded	$\xrightarrow{A}$	
	$\xleftarrow{g}$	Lookup key A in the database to get g and h
Choose $r \in_R \mathbb{Z}_g$, let $a = g^r$	$\xrightarrow{a}$	
	$\xleftarrow{b}$	Choose $b \in_R \mathbb{Z}_g$
Let $c = r + f(\tilde{V}) \cdot b$	$\xrightarrow{c}$	
Report decision	$\xleftarrow{ah^b \stackrel{?}{=} g^c}$	

Where f is some fuzzy extractor

Zero Knowledge Protocol

- Correctness: $ah^b = g^r\left(g^{f(V)}\right)^b = g^{r+f(V)\cdot b} \stackrel{?}{=} g^{r+f(\tilde{V})\cdot b} = g^c$

Zero Knowledge Protocol

- Correctness: $ah^b = g^r\left(g^{f(V)}\right)^b = g^{r+f(V)\cdot b} \stackrel{?}{=} g^{r+f(\tilde{V})\cdot b} = g^c$
  - This will hold if $f(V) = f(\tilde{V})$

Zero Knowledge Protocol

- Correctness: $ah^b = g^r(g^{f(V)})^b = g^{r+f(V)\cdot b} \stackrel{?}{=} g^{r+f(\tilde{V})\cdot b} = g^c$
  - This will hold if $f(V) = f(\tilde{V})$
- Security reduces to discrete log

Zero Knowledge Protocol

- Correctness: $ah^b = g^r(g^{f(V)})^b = g^{r+f(V)\cdot b} \stackrel{?}{=} g^{r+f(\tilde{V})\cdot b} = g^c$
  - This will hold if $f(V) = f(\tilde{V})$
- Security reduces to discrete log
- ...But can the fuzzy extractor accurately authenticate someone?

Challenges and Limitations

- Zero knowledge means information loss

Challenges and Limitations

- Zero knowledge means information loss
- The protocol is only as good as the fuzzy extractor

Challenges and Limitations

- Zero knowledge means information loss
- The protocol is only as good as the fuzzy extractor
  - Accuracy vs Precision tradeoff

Challenges and Limitations

- Zero knowledge means information loss
- The protocol is only as good as the fuzzy extractor
  - Accuracy vs Precision tradeoff
  - User expects to be authenticated first try

Challenges and Limitations

- Zero knowledge means information loss
- The protocol is only as good as the fuzzy extractor
  - Accuracy vs Precision tradeoff
  - User expects to be authenticated first try
  - Precision varies by user

Challenges and Limitations

- Zero knowledge means information loss
- The protocol is only as good as the fuzzy extractor
  - Accuracy vs Precision tradeoff
  - User expects to be authenticated first try
  - Precision varies by user
- Conclusion: performance of machine learning approach is likely not attainable with zero-knowledge

Demo

References

Francesco Bergadano, Daniele Gunetti, and Claudia Picardi.
User authentication through keystroke dynamics.
ACM Trans. Inf. Syst. Secur., 5(4):367–397, November 2002.

S. Bleha, C. Slivinsky, and B. Hussien.
Computer-access security systems using keystroke dynamics.
IEEE Trans. Pattern Anal. Mach. Intell., 12(12):1217–1222, December 1990.

Soumik Mondal, Patrick Bours, and S. Z. Syed Idrus.
Complexity measurement of a password for keystroke dynamics: Preliminary study.
In *Proceedings of the 6th International Conference on Security of Information and Networks*, SIN '13, pages 301–305, New York, NY, USA, 2013. ACM.

Fabian Monrose and Aviel D. Rubin.
Authentication via keystroke dynamics.
In *ACM Conference on Computer and Communications Security'97*, pages 48–56, 1997.

Deian Stefan and Danfeng Yao.
Keystroke-dynamics authentication against synthetic forgeries.
In *CollaborateCom*, pages 1–8. IEEE, 2010.